Photonic Network Communications, 9:3, 311–335, 2005

A Detailed Analysis and Performance Comparison of Wavelength Reservation Schemes for Optical Burst Switched Networks†

Jing Teng, George N. Rouskas*
Department of Computer Science, North Carolina State University, USA
E-mail: {jteng, rouskas}@eos.ncsu.edu

Received January 20, 2004; Revised January 26, 2004; Accepted January 28, 2004

Abstract. We present a detailed analysis of the JIT, JET, and Horizon wavelength reservation schemes for optical burst switched (OBS) networks. Our analysis accounts for several important parameters, including the burst offset length, and the optical switching and hardware processing overheads associated with bursts as they travel across the network. The contributions of our work include: (i) analytical models of JET and Horizon (on a single OBS node) that are more accurate than previously published ones, and which are valid for general burst length and offset length distributions; (ii) the determination of the regions of parameter values in which a more complex reservation scheme reduces to a simpler one; and (iii) a new reservation scheme, JIT^+, which is as simple to implement as JIT, but whose performance tracks that of Horizon and JET. We compare the performance of the four wavelength reservation schemes on a single OBS node, as well as on a path of OBS nodes with cross traffic, under various sets of parameter values. Our major finding is that, under reasonable assumptions regarding the current and future state-of-the-art in optical switch and electronic hardware technologies, the simplicity of JIT and JIT^+ seem to outweigh any performance benefits of Horizon and JET.

Keywords: optical burst switching, wavelength reservation, just-in-time signaling

1 Introduction

Optical burst switching (OBS) is a technology positioned between wavelength routing (i.e., circuit switching) and optical packet switching. All-optical circuits tend to be inefficient for traffic that has not been groomed or statistically multiplexed, and optical packet switching requires practical, cost-effective, and scalable implementations of optical buffering and optical header processing, which are several years away. OBS is a technical compromise that does not require optical buffering or packet-level parsing, and it is more efficient than circuit switching when the sustained traffic volume does not consume a full wavelength. The transmission of each burst is preceded by the transmission of a setup (also referred to as burst header control) message, whose purpose 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. An OBS source node does not wait for confirmation that an end-to-end connection has been set-up; instead it starts transmitting a data burst after a delay (referred to as *offset*), following the transmission of the setup message. We assume that OBS nodes have no buffers, therefore, in case of congestion or output port conflict, they may drop bursts.

OBS networks have received considerable attention recently, mainly through theoretical investigations. A number of wavelength reservation schemes have been proposed for OBS, including just-enough-time (JET) [1], Horizon [2], just-in-time (JIT) [3,4], and wavelength-routed OBS [5] which uses two-way reservations. The burst loss performance of OBS networks has been studied extensively using either simulation or simple analytical models [2,6–10]. Typically, an output port of an OBS node has been analyzed

*Corresponding author.
† This work was supported by MCNC-RDI as part of the Jumpstart project.

assuming Poisson arrivals and no buffering [7–9]. Under these assumptions, an output port can be modeled by a finite number of servers, each representing a wavelength, with no queue. Then, the probability that a burst destined to this output port is lost can be obtained from the Erlang-B formula. An output port can also be modeled as an $M/M/m/K$ queue by assuming Poisson arrivals and buffering [2,10], where m is the number of wavelengths and K−m is the capacity of the buffer. A similar model that accounts for multiple classes of bursts, each class characterized by a different offset length, was developed in [6]. Other issues related to OBS networks that have been investigated in the literature include control architectures [11,12], wavelength scheduling algorithms [13,14], the effect of optical buffers [15], burst assembly [16,17] and traffic shaping [9] at the edge of the network, and quality of service (QoS) support [8,10].

Whereas all the above studies of OBS are theoretical in nature, we have been collaborating with MCNC-RDI since late 2000 to build a proof-of-concept OBS implementation under the ARDA-funded Jumpstart project [18]. (ARDA focuses on high-performance data communications requirements that cannot be addressed by technologies used in today's Internet [19].) We have developed an open, published specification of the Jumpstart JIT signaling protocol [4,20], inspired by an earlier work by Wei and McFarland [3]. The JIT protocol is significantly simpler than either JET or Horizon, since it does not involve complex scheduling or void filling algorithms; therefore, it is amenable to hardware implementation. MCNC-RDI has developed JIT protocol acceleration card (JIT-PAC) network controllers which implement the signaling protocol in FPGA, and deployed them at three ATDNet sites in November 2002 for experimentation and testing [21]. This is the first OBS field trial known to us.

While JIT is conceptually simple, previous studies have shown that JIT performs worse than either JET or Horizon in terms of burst loss probability. Indeed, given the sophisticated scheduling and void filling algorithms that JET and Horizon require, the fact that these schemes should outperform JIT might seem a reasonable one at first thought. However, most of the existing studies ignore many important parameters such as the offset length, the processing time of setup messages, and the optical switch configuration time, which have significant impact on burst loss probability. For instance, it is not unreasonable to assume that, due to complex operations and/or large number of memory lookups, the processing of setup messages under JET or Horizon will take longer than under JIT; in this case it is not clear whether the more efficient scheduling of JET and Horizon will outweigh the higher processing overhead incurred. Similarly, if the optical switch configuration time is much longer than the mean burst length, any differences in scheduling efficiency will have little effect on overall burst loss probability. Therefore, there is a need for more detailed studies in order to explore in depth the differences among the various wavelength reservation schemes, and to establish the regions of network operation where one scheme may outperform the others.

In this paper, we develop accurate models for an OBS node operating under the JET, JIT, and Horizon wavelength reservation schemes. The analytical models assume Poisson arrivals, but are valid for arbitrary burst length distributions and arbitrary offset length distributions. The models also account for the processing time of setup messages and the optical switch configuration times, and thus, are very general. One important finding of our work is that, under reasonable assumptions regarding current and future capabilities of optical switch and electronic (hardware) processing technologies, the performance in terms of burst drop probability of the (significantly simpler) JIT reservation scheme is very similar to that of the more complex JET or Horizon schemes. For network scenarios where JET or Horizon outperform JIT, we introduce JIT$^+$, a new reservation scheme which retains the simplicity of JIT but exhibits a performance behavior close to JET and Horizon. Another contribution made possible by our analysis is the characterization of the regions of network operation in which a more complex reservation scheme reduces to a simpler one (i.e., when JET reduces to Horizon, Horizon to JIT$^+$, or JIT$^+$ to JIT).

This paper is organized as follows. Section 2 describes the OBS network we consider in this study, and introduces important system parameters used in our analysis. Section 3 provides a detailed description of the JIT, JET, and Horizon wavelength reservation schemes, discusses issues related to their hardware implementation, and introduces a new reservation scheme called JIT^+. In Section 4, we develop analytical models of a single OBS node that capture the performance of the four reservation schemes. In Section 5, we present numerical results to compare the relative performance of the four schemes, both on a single OBS node and a path of OBS nodes with cross-traffic, under a wide range of system parameter values that correspond to current and projected technology. We then conclude the paper in Section 6.

2 The OBS Network Under Study

We consider a network consisting of OBS nodes interconnected by bidirectional fiber links, as shown in Fig. 1. Users are attached to edge switches of the OBS network, also using bidirectional fiber links. We assume that all fiber links, including links between switches as well as links between a user and an edge switch, support the same set of $W + 1$ wavelengths in each direction. One wavelength is used for signaling (i.e., it carries setup messages) and the other W wavelengths carry data bursts.

Consider an OBS node in the network, and let P denote the number of input and output ports of the node. Each (input or output) port is attached to a fiber link connecting the node to other OBS nodes in the network or to burst-transmitting users. The OBS node consists of two main components, as illustrated in Fig. 1:

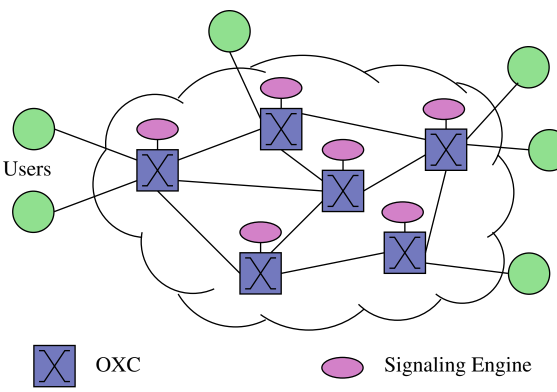

Fig. 1 An OBS network.

1. A *signaling engine*, which implements the OBS signaling protocol and related forwarding and control functions. To avoid bottlenecks in the control plane and to achieve operation at wire speeds, we assume that the signaling engine is implemented in hardware. (For example, the JITPAC hardware [21], which was developed by MCNC-RDI, implements the JIT signaling engine in FPGA.)
2. An *optical cross-connect* (OXC), which performs the switching of bursts from input to output. We assume that the OXC consists of a non-blocking space-division switch fabric, with no optical buffers. We also assume that the OXC has full conversion capability, so that an optical signal on any wavelength at any input port can be converted to any wavelength at any output port.

The OBS node does not employ any optical buffers (e.g., fiber delay lines). Consequently, bursts that cannot be switched are dropped.

Whereas burst wavelengths are optically switched at the OBS node, the signaling wavelength is terminated at the node, the information it carries is converted to electronic form, and the resulting signal is passed to the signaling engine. The signaling engine decodes the electronic signal and processes each incoming message using the appropriate rules (i.e., finite state machines [22] of the JIT protocol). Processing a signaling message may involve one or more actions, including: (1) the determination of a next hop switch for a burst; (2) the forwarding of signaling messages to upstream or downstream nodes; (3) the configuration of the OXC switching elements to optically switch bursts from an input to an output port; and (4) the handling of exception conditions.

The following parameters play an important role in the performance of the OBS node, and will be used in our analysis.

- T_{OXC} is the amount of time it takes the OXC to configure its switch fabric to set up a connection from an input port to an output port. In other words, T_{OXC} is the delay incurred between the instant the OXC receives a command from the signaling engine to set up a connection from an input port to an output port, until the instant the appropriate path within the optical switch is complete and can be used to switch a burst. This delay includes the configuration of optical switch elements within the OXC, e.g., the raising of a micro-mirror in the case of a MEMS switch. In this study, we assume that this configuration delay is largely independent of the pair of input/output ports that must be connected, as well as of the state of the optical switch at the time the connection must be performed; this assumption is valid for optical switch technologies under development, including MEMS mirror arrays [23]. Therefore, we take T_{OXC} as a constant in our study.

- $T_{\mathrm{setup}(X)}$ is the amount of time it takes an OBS node to process the setup message under reservation scheme X, where X can be any of JIT, JET, Horizon, or JIT$^+$. Since, as we explain in Section 3, different reservation schemes have different processing and scheduling requirements, this amount of time is a function of the reservation scheme employed. However, for a given scheme X, we assume that $T_{\mathrm{setup}(X)}$ is constant across all bursts. This is a reasonable assumption since processing of signaling messages will most likely be performed in hardware, as we have demonstrated in the Jumpstart project [22], and thus, the processing time can be bounded.
- $T_{\mathrm{offset}(X)}$ is the offset value of a burst under reservation scheme X. The offset value depends on (1) the wavelength reservation scheme, (2) the number of nodes the burst has already traversed (since the offset value decreases as the burst travels further into the network), and (3) other factors, such as whether the offset is used for service differentiation [10]. The primary consideration in the calculation of the offset value is to ensure that the first bit of the burst arrives at the destination node shortly after this node is ready to receive it (i.e., just after the destination has processed the setup message announcing the burst). The delay between the setup message and the first bit of the burst shrinks as the two propagate along the path to the destination. This is because the setup message encounters processing delays at each OBS node in the path, whereas the burst travels transparently in the optical domain. In addition, one must account for the switch setup delay T_{OXC} of the last OXC in the path.

Let k be the number of OBS nodes in the path of a burst from source to destination. Based on the above observations, it is easy to see that the minimum offset value to guarantee that the burst will arrive at the destination immediately after the setup message has been processed is equal to:

$$T_{\mathrm{offset}}^{(\mathrm{min})}(X) = kT_{\mathrm{setup}}(X) + T_{\mathrm{OXC}} \tag{1}$$

We note that the actual offset length can take any value larger than the minimum one shown in the above expression; in fact, the models we develop later can account for offset lengths of arbitrary distributions.

3 Wavelength Reservation Schemes for OBS Nodes

The manner in which output wavelengths are reserved for bursts is one of the principal differentiating factors among OBS variants. We distinguish between two types of reservations: *immediate* and *delayed*. For simplicity, in the following we will use the notation T_{offset} and T_{setup} without specifying the reservation scheme X, whenever the latter is obvious from the context.

3.1 Immediate Reservation (JIT)

Immediate reservation, exemplified by the JIT family of OBS protocols [3,4], works as follows:

> an output wavelength is reserved for a burst immediately after the arrival of the corresponding *setup* message; if a wavelength cannot be reserved at that time, then the*setup* message is rejected and the corresponding burst is dropped.

We illustrate the operation of JIT in Fig. 2. Let t be the time a setup message arrives at some OBS node along the path to the destination user; this node can be any of the "ingress," "intermediate," or "egress" switches in the figure. As the figure

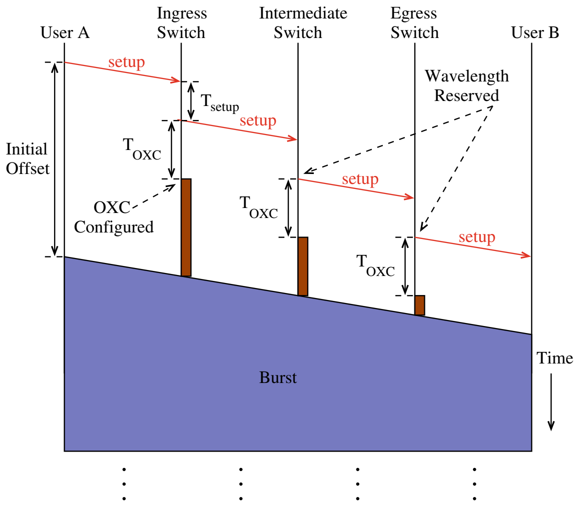

Fig. 2. Immediate wavelength reservation.

shows, once the processing of the setup message is complete at time $t + T_{\text{setup}}$, a wavelength is immediately reserved for the upcoming burst, and the operation to configure the OXC fabric to switch the burst is initiated. When this operation completes at time $t + T_{\text{setup}} + T_{\text{OXC}}$, the OXC is ready to carry the burst.

Note that, by the offset definition, the burst will not arrive at the OBS node under consideration until time $t + T_{\text{offset}}$. As a result, the wavelength allocated to the burst remains idle for a period of time equal to $(T_{\text{offset}} - T_{\text{setup}} - T_{\text{OXC}})$. We also note that the offset value decreases along the path to the destination. Consequently, as the figure shows, the deeper inside the network an OBS node is located, the shorter the idle time between the instant the OXC has been configured and the arrival of the burst.

Fig. 3 offers another perspective on how immediate reservation works, by considering the operation of a single output wavelength of an OBS node. Each such wavelength can be in one of two states: reserved or free. Fig. 3 shows two successive bursts, i and $i + 1$, *successfully transmitted* on the same output wavelength; the figure does not show any dropped bursts that may have arrived between the two successful bursts.

As we can see in Fig. 3, the setup message corresponding to the ith burst arrives at the switch at time t_1, when we assume that the wavelength is free. This message is accepted by the switch, the status of the wavelength becomes reserved and, after an amount of time equal to the offset, the first bit of the optical burst arrives at the switch at time t_2. The last bit of the burst arrives at the switch at time t_3, at which instant the status of the wavelength is updated to free. Note that, any new setup message that arrives between t_1 and t_3 when the status of the wavelength is reserved is rejected by the switch, since the wavelength cannot be immediately reserved for the new burst. The length of the interval, $t_3 - t_1$, during which new setup messages are rejected, is equal to the sum of the offset value and the length of burst i.

Suppose now that the next setup message for this wavelength arrives at the switch at time $t_4 > t_3$, while the wavelength is still free. Consequently, the burst corresponding to this message becomes the $(i+1)$th burst to *successfully depart* on this wavelength; note that this burst may not be the $(i+1)$th *arriving* burst, since some setup message(s) may have been rejected by the switch before time t_3. After an amount of time equal to the offset, the burst arrives at time t_5, and its

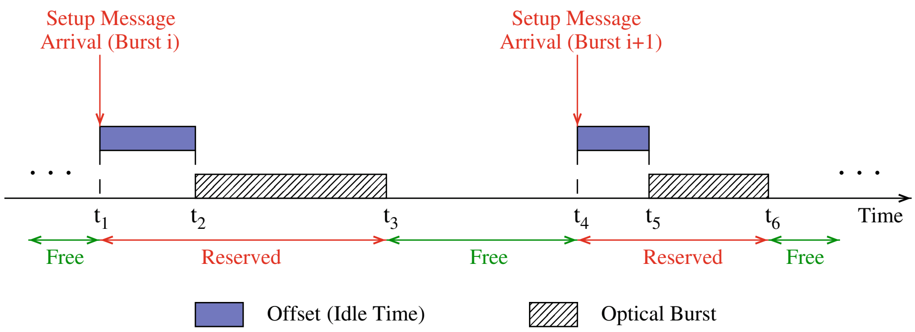

Fig. 3. Operation and departure process of a wavelength with immediate reservation (JIT).

transmission ends at time t_6, at which instant the wavelength becomes free again.

As Fig. 3 illustrates, the operation of a wavelength with immediate reservation is conceptually simple. Time on the wavelength is divided into periods during which the wavelength is reserved, followed by periods during which it is free. The length of a reserved period is equal to the burst length plus the corresponding offset, while the length of a free period is equal to the time until the arrival of the next setup message. Also, service on each wavelength is first-come, first-served (FCFS), in the sense that bursts are served in the order in which their corresponding setup messages arrive at the switch.

3.2 Delayed Reservation

The Horizon [2] and JET [1,24] protocols employ a delayed reservation scheme which operates as follows:

> an output wavelength is reserved for a burst just before the arrival of the first bit of the burst; if, upon arrival of the *setup* message, it is determined that no wavelength can be reserved at the appropriate time, then the *setup* message is rejected and the corresponding burst is dropped.

Fig. 4 illustrates the operation of delayed reservation. Let us again assume that a setup message arrives at an OBS node at time t, in which case the first bit of the corresponding burst is expected to arrive at time $t + T_{\text{offset}}$. Assuming that the burst can be accepted, the setup message reserves a wavelength for the burst starting at time $t' = t + T_{\text{offset}} - T_{\text{OXC}}$. As shown in the figure, at time t', the OBS node instructs its OXC fabric to configure its switch elements to carry the burst, and this operation completes just before the arrival of the first bit of the burst. Thus, whereas immediate reservation protocols only permit a single outstanding reservation for each output wavelength, delayed reservation schemes allow multiple setup messages to make future reservations on a given wavelength (provided of course, that these reservations, i.e., the corresponding bursts, do not overlap in time). We also note that, when a burst is accepted, the output wavelength is reserved for an amount of time equal to the length of the burst *plus* T_{OXC}, in order to account for the OXC configuration time.

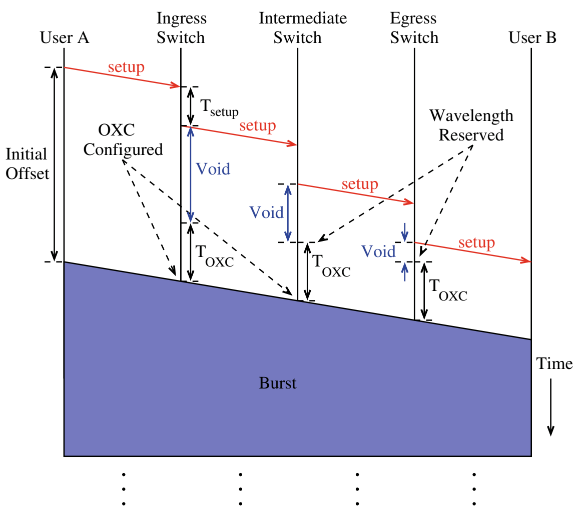

Fig. 4. Delayed reservation.

As we can see in Fig. 4, a *void* is created on the output wavelength between time $t + T_{\text{setup}}$, when the reservation operation for the upcoming burst is completed, and time $t' = t + T_{\text{offset}} - T_{\text{OXC}}$, when the output wavelength is actually reserved for the burst. If the offset value T_{offset} is equal to the minimum value in Expression (1), then the length of this void at some OBS node x is equal to rT_{setup}, where r is the number of OBS nodes in the path from x to the destination of the burst. Consequently, the void created by a given burst decreases in size as the burst travels along its path.

Delayed reservation schemes can be further classified according to whether or not they employ specialized burst scheduling algorithms in an attempt to make use of the voids created by earlier setup messages, by transmitting bursts whose setup messages arrive later. Usually, such scheduling techniques are referred to as *void filling* algorithms.

3.2.1 Delayed Reservation Without Void Filling (Horizon)

Delayed reservation schemes, such as Horizon [2], that do not perform any void filling, are typically less complex than schemes with void filling, such as JET. The Horizon scheme takes its name from the fact that each wavelength is associated with a *time horizon* for burst reservation purposes. This time horizon is defined as “the earliest time after which there is no planned use of the channel (wavelength)”. Under this scheme,

> an output wavelength is reserved for a burst only if the arrival time of the burst is later than the time horizon of the wavelength; if, upon arrival of the setup message, it is determined that the arrival time of the burst is earlier than the smallest time horizon of any wavelength, then thesetup message is rejected and the corresponding burst dropped.

When a burst is scheduled on a given wavelength, then the time horizon of the wavelength is updated to the departure instant of the burst *plus* the OXC configuration time T_{OXC}. Consequently, under Horizon, a new burst can be scheduled on a wavelength only if the first bit of the burst arrives *after* all currently scheduled bursts on this wavelength have departed.

Fig. 5 shows two bursts transmitted successively on a given wavelength out of an OBS node using the Horizon reservation scheme. The setup message of burst *i* arrives at the OBS node at time t_1, and the last bit of this burst leaves the node at time t_4. Since the OXC needs an amount of time equal to T_{OXC} to reconfigure its switching elements to perform a connection from another input port to this output wavelength, no new bursts can be scheduled on this wavelength until time $t_5 = t_4 + T_{OXC}$. Therefore, at time t_1, i.e., when burst *i* is accepted, t_5 becomes the time horizon of this channel.

Let us now suppose that, as Fig. 5 illustrates, the setup message of burst $i+1$ arrives at the OBS node at time $t_2 > t_1$. The node uses the offset length information carried in the setup message to calculate that the first bit of this burst will arrive at time t_6. Since $t_6 > t_5$, burst $i+1$ is scheduled for transmission on this wavelength, and the time horizon is updated accordingly to $t_7 + T_{OXC}$, where t_7 is the instant the transmission of burst $i+1$ ends. This example shows that the offset of a burst (in this case, burst $i+1$) may overlap with the offset and/or transmission of another burst (i.e., burst *i*). However, bursts are scheduled in a strict FCFS manner determined by the order of arrival of their respective setup messages.

3.2.2 Delayed Reservation With Void Filling (JET)

JET [24] is the best known delayed wavelength reservation scheme that uses void filling. Under JET,

> an output wavelength is reserved for a burst if the arrival time of the burst (1) is later than the time horizon of the wavelength, or (2) coincides with a void on the wavelength, and the end of the burst (plus the OXC configuration time T_{OXC}) occurs before the end of the void; if, upon arrival of the *setup* message, it is determined that none of these conditions are satisfied for any wavelength, then the *setup* message is rejected and the corresponding burst dropped.

Note that, bursts which are accepted because their arrival and departure instants satisfy condition (2) above would have been rejected by an OBS node using Horizon. Consequently, JET is expected to perform better than Horizon in terms of burst drop probability. On the other hand, the void filling algorithm must keep track of, and search, the starting and ending times of all voids on the various wavelengths, resulting in a more complex implementation than either Horizon or JIT; a more detailed discussion of implementation issues is provided in Section 3.3.

Fig. 6 illustrates the void-filling operation of JET. The figure shows two bursts, *A* and *B*, which

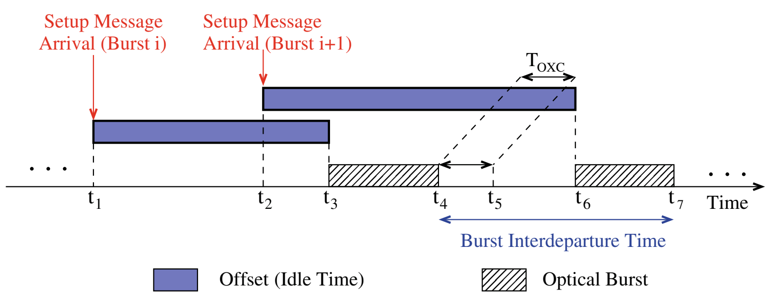

Fig. 5. Departure process of a wavelength with delayed reservation and no void filling (Horizon).

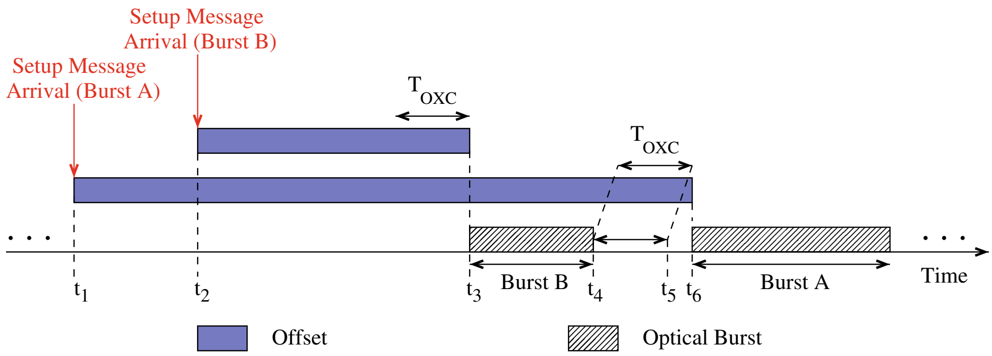

Fig. 6. Non-FCFS service of a wavelength in an OBS node with delayed reservation and void filling (JET).

are both transmitted on the same output wavelength. The setup message for burst *A* arrives first, followed by the setup message for burst *B*. As we show in the figure, burst *A* has a long offset. Upon receipt of its setup message, the switch notes the later arrival of burst *A*, but does not initiate any connection within its cross-connect fabric. Once burst *A* has been accepted, a void is created, which is the interval of time until the arrival of the first bit of the burst at time t_6. Let us assume that at time t_2 when the setup message for burst *B* arrives, no other burst transmissions have been scheduled within this void.

Upon the arrival of the setup message for burst *B* at time t_2, the switch notes that burst *B* will arrive *before* the arrival of burst *A*, and runs a void filling algorithm [11,13] to determine whether it can accept the new burst. In order to accept the new burst, there must be sufficient time between the end of the transmission of burst *B* and the arrival of burst *A* for the switch to reconfigure its cross-connect fabric to accommodate burst *A*. For the scenario depicted in Fig. 6, burst *B* is accepted, and it completes service before the arrival of the first bit of burst *A*. Since the setup message for burst *B* arrived after the setup message for burst *A*, this operation results in a non-FCFS service of bursts.

3.3 Implementation Considerations

Let us now consider the amount of state information that the OBS node needs to maintain for each output port in order to implement each of the JIT, JET, and Horizon schemes, as well as the running time complexity of the corresponding burst scheduling algorithms. We distinguish between two types of state information: information that is necessary to perform OXC configuration operations, and information needed for the burst scheduling algorithm. We also note that memory access operations dominate the execution time in a hardware implementation of a protocol, and thus, we will focus on the memory access requirements of the three reservation schemes.

Let us first consider JIT. As Fig. 3 illustrates, an output wavelength can be either free or reserved, and while it is reserved, no new bursts can be accepted for transmission. Therefore, for OXC configuration purposes, an OBS node only needs to maintain a *wavelength vector* of size W for each output port, where W is the number of wavelengths per fiber (port). When wavelength w is reserved for a burst, field w of the vector is set to the time the burst transmission will complete; at that time, the wavelength is freed by setting the field w to a special value. The same vector can be used for burst scheduling. Since it makes no difference which wavelength carries a particular burst, the OBS node may simply reserve the first free wavelength indicated by the wavelength vector. Alternatively, the OBS node may return the first free wavelength following the wavelength that was reserved last (in order to balance the burst load across the various wavelengths), or it could first check whether the incoming wavelength of the burst is available (to avoid conversion). All these operations take constant time and require only a single memory lookup, hence JIT is well-suited to hardware implementation [4,21].

Now let us consider Horizon. Horizon allows multiple outstanding reservations for each output wavelength, therefore, an OBS node needs to maintain W *reservation lists* per output port, one for each wavelength. A reservation list consists of fields indicating the start and end time of each burst reservation on a particular wavelength, and is used by the OBS node to configure its OXC. For scheduling purposes, the OBS node must maintain the time horizon (i.e., the end of the latest reservation) for each wavelength, as well as a list of the time horizons in increasing order [2,13]. When a setup message arrives, the Horizon algorithm reserves the wavelength with the latest time horizon that is earlier than the arrival of the corresponding burst. This algorithm takes $O(W)$ time to schedule each burst, and also requires a large number of memory lookup/write operations: one operation to update the reservation list (since a new reservation is always appended at the end of a list), and $O(W)$ operations to update the ordered list of time horizons.

Similar to Horizon, the JET reservation scheme requires the OBS node to maintain one reservation list per wavelength for each output port. However, adding a new reservation requires a traversal of the list to insert the reservation at the correct place (i.e., void), hence this operation is much more expensive than in Horizon in terms of memory access. The cost of a burst scheduling operation depends on the actual scheduling algorithm used. The LAUC-VF (latest available unused channel with void filing) algorithm proposed in [11] requires a sequential search of *all* wavelength reservation lists for each burst; this takes time $O(m)$ [13], where m is the number of voids, which can be larger than W. Hence, this algorithm is expensive in terms of running time and number of memory accesses for hardware implementation. A faster algorithm was proposed recently in [13] which only takes time $O(\log m)$. However, this algorithm requires the OBS node to maintain complex data structures such as red-black trees; therefore, this algorithm is better suited for software, rather than hardware, implementation.

3.4 Modified Immediate Reservation (JIT$^+$)

Based on the above discussion regarding the relative complexity of the JIT, JET, and Horizon reservation schemes, as well as our observations regarding their relative performance under a wide range of values for the various system parameters (refer to Section 5), we now present a new reservation scheme, which we refer to as JIT$^+$. More specifically, JIT$^+$ operates as follows:

> an output wavelength is reserved for a burst if (1) the arrival time of the burst is later than the time horizon of the wavelength and (2) the wavelength has *at most* one other reservation.

JIT$^+$ does not perform any void filling. JIT$^+$ attempts to improve upon JIT by making a delayed burst reservation on a wavelength, even when the wavelength is currently reserved by another burst. However, whereas Horizon and JET permit an unlimited number of delayed reservations per wavelength, JIT$^+$ limits the number of such operations to at most one per wavelength.

Fig. 7 illustrates the operation of JIT$^+$ by considering three bursts, i, $i+1$, and $i+2$; note that the arrival times, offsets, and lengths of bursts i and $i+1$ are identical to those in Fig. 5. As in Fig. 5, when the setup message for burst i arrives at time t_1, the burst is accepted, the wavelength is reserved, and the time horizon is updated to t_5.

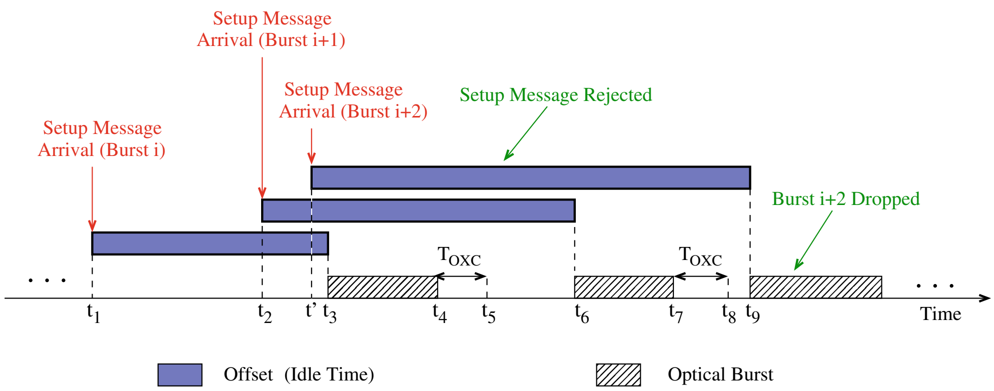

Fig. 7. Operation of the modified immediate reservation scheme (JIT$^+$).

When the setup message for burst $i+1$ arrives at time t_2, as in Horizon, the burst is accepted and the time horizon is updated to t_8; note that this burst would have been dropped by JIT. At this time, the wavelength has one outstanding reservation, the one for burst $i+1$. Consequently, when the setup message for burst $i+2$ arrives at time t', the setup message is rejected and the corresponding burst dropped; note that this burst would have been accepted by Horizon, since the first bit of burst $i+2$ is expected to arrive at time $t_9 > t_8$, where t_8 is the current time horizon. In fact, no more bursts will be scheduled on this wavelength until after the departure of burst i at time t_4.

Since each wavelength may be reserved for at most two bursts, to implement JIT$^+$, an OBS node needs to maintain a wavelength vector with W fields for each output port, where W is the number of wavelengths. Each field w, corresponding to wavelength w, consists of two values, namely, the departure instants of each of the two bursts that may be scheduled on the wavelength. Updating the field for wavelength w (e.g., when a new burst is reserved or when one departs) takes constant time and requires a single memory access operation. To avoid the $O(W)$ sorting operations on the time horizons of the various wavelengths required by the LAUC algorithm employed in the Horizon scheme, JIT$^+$ reserves the first wavelength that can accommodate a burst; alternatively it may return the first available wavelength following the wavelength that was reserved last, or, in order to avoid wavelength conversion, it may first check whether the incoming wavelength of the burst is available. All these operations take constant time, and require a single memory lookup, hence JIT$^+$ maintains all the advantages of JIT in terms of simplicity of hardware implementation.

4 Models of an OBS Node

In this section, we develop three analytical models for an output port p of an OBS node, one for each of the three reservation schemes JIT, JET, and Horizon. In our analysis, we make the following assumptions:

- setup messages corresponding to bursts destined to output port p arrive at the OBS node according to a Poisson process with rate λ; this arrival rate is the total rate over all input ports. The assumption of Poisson arrivals is made mainly for mathematical tractability, and is common in the OBS literature [2,6–10].
- Burst lengths follow a general distribution with CDF $B(l)$ and Laplace transform $B^\star(s)$. We let $1/\mu$ denote the mean of the burst length distribution.
- Offset lengths follow a general distribution with CDF $G(z)$ and Laplace transform $G^\star(s)$. We also let $\overline{T}_{\text{offset}}(X)$ denote the mean offset length under reservation scheme X.
- An output wavelength is reserved for a given burst for a period of time that is *larger than* the length of the burst; at a minimum, the wavelength must be reserved for the duration of the burst length *plus* the OXC configuration time T_{OXC}, to allow for setting up the optical switch fabric to establish a connection from the input to the output port. Therefore, we define the *effective service time* of a burst as the amount of time that an output wavelength is reserved for the burst. As we shall see, the effective service time of the burst depends on the wavelength reservation scheme used.

We note that, while the burst arrival rate λ and the burst length distribution are *not* affected by the reservation scheme (JIT, JET, Horizon, or JIT$^+$), because of (1), the offset length distribution is affected by the choice of reservation scheme.

Note that we have assumed that setup messages arrive as a Poisson process with rate λ. Let us now concentrate on the arrival process of the corresponding bursts, rather than that of the setup messages. The arrival time of a burst is the arrival time t of its setup message plus an offset, which is distributed according to a general distribution $G(z)$. One way of thinking about this burst arrival process is to assume that bursts arrive *at the same time* as their corresponding setup messages (i.e., as a Poisson process with rate λ), but they have to be served by a *fictitious* infinite server (i.e., an $M/G/\infty$ queue) before they enter the OBS node, as shown in Fig. 8. The service time at this infinite server is distributed according to the CDF of the offset length, $G(z)$. As a result, the actual arrival of a burst to the OBS node is indeed the arrival time of its setup message plus an offset time distributed

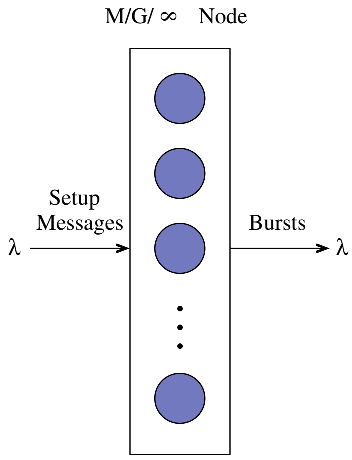

Fig. 8. Burst arrival process.

according to CDF $G(z)$. It is well-known that the departure process of an $M/G/\infty$ queue is a Poisson process with rate λ, the same as the arrival process. Therefore, burst arrivals to the OBS node are also Poisson with rate λ.

We note that the above $M/G/\infty$ model assumes *optimal* scheduling and void filling algorithms, in the sense that no burst is dropped if it can be carried by the switch; in practice, fast suboptimal algorithms may be used, in which case some bursts may be dropped even if they would be scheduled under an optimal algorithm. Furthermore, the $M/G/\infty$ model is an approximation since the underlying assumption is that the decision to accept or drop the burst is taken *at the moment the first bit of the burst arrives*. In other words, this model is exact only under the assumption that processing of setup messages and the OXC configuration takes zero time. In reality, the decision to accept or drop a burst is taken at the instant its setup message arrives, and if a setup message is rejected then the corresponding burst never arrives at the OBS node, resulting in a non-Poisson arrival process for bursts. However, the $M/G/\infty$ model is both conceptually simple and reasonably accurate, and we will make use of it in the analysis of some of the reservation schemes.

We model the output port of an OBS node as a multiple server loss system, and we use the Erlang-B formula to obtain the burst drop probability. The Erlang-B formula for an m-server loss system with traffic intensity ρ is given by

$$Erl(\rho, m) = \frac{\rho^m/m!}{\sum_{i=0}^{m} \rho^i/i!} \tag{2}$$

In the following subsections, we determine accurate values for the intensity ρ under each reservation scheme. Since the loss probability in an m-server loss system is insensitive to the service time distribution, we use the Erlang-B formula above for any distribution of the effective service time of bursts.

4.1 A Model of JIT

In order to determine the effective service time of a burst under the JIT reservation scheme, let us refer again to Fig. 3. We observe that, for a given burst, a wavelength is reserved for a length of time that is equal to the sum of two time periods. The duration of the first period is equal to the burst offset, and is distributed according to CDF $G(z)$ with a mean $\overline{T}_{\text{offset}}(\text{JIT})$. The duration of the second period is equal to the burst length, and is distributed according to CDF $B(l)$ with a mean $1/\mu$. Consequently, the Laplace transform of the distribution of the effective service time of bursts is given by $G^\star(s)B^\star(s)$, with mean $1/\mu + \overline{T}_{\text{offset}}(\text{JIT})$.

Based on these observations, an output port of an OBS node using JIT behaves as an $M/G/W/W$ loss system, where W is the number of wavelengths of the port. The traffic intensity ρJIT of the queue is:

$$\rho(\text{JIT}) = \lambda\left(\frac{1}{\mu} + \overline{T}_{\text{offset}}(\text{JIT})\right) \tag{3}$$

and the burst drop probability is given by $Erl(\rho(\text{JIT}), W)$. We also note that, under the assumption that setup messages arrive as a Poisson process, the $M/G/W/W$ queue is an exact model for JIT. This model has been used in earlier studies, e.g., in [7], where, however, the assumption was made that burst (rather than setup message) arrivals are Poisson; in that case, the model is only approximate.

4.2 A Model of JET

The operation of an OBS node under the delayed reservation scheme is more complicated than under immediate reservation (i.e., JIT). Let us first consider the case in which void filling is employed [11,13] when allocating a wavelength to a burst, as in the JET [24] reservation scheme. The difficulty in this case arises from two observations regarding burst transmissions on a given output wavelength. First, the offset of a given burst may overlap with the offset and/or transmission of one or more other bursts. Second, bursts are not necessarily served in an FCFS fashion. This overlap feature and resulting non-FCFS service were illustrated in Fig. 6.

To overcome the difficulty introduced by the offset overlap and the non-FCFS service, let us concentrate on the departure process of a given output wavelength. In Fig. 9, we show two bursts transmitted successively out of the switch on a given wavelength. We number the bursts in the order in which they *depart* the switch, so that burst $i+1$ is the first burst to be transmitted out on this wavelength after burst i; note that, due to the possibility for void filling, this may not be the order in which the setup messages of the two bursts arrived.

As Fig. 9 illustrates, the first bit of burst i arrives at the OBS node at time t_1, and the last bit of the same burst leaves the switch at time t_2. Recall that the OXC needs an amount of time equal to T_{OXC} to reconfigure its switching elements to perform a connection from another input port to this output wavelength. Therefore, the switch cannot accommodate a new burst on this wavelength until time t_3, which is such that $t_3 = t_2 + T_{\mathrm{OXC}}$. In fact, any setup message for a burst scheduled to arrive at the switch in the time interval between t_2 and t_3 would have been rejected by the switch scheduling algorithm. Therefore, we can think of a burst as occupying the channel not only during its transmission time (equal to its length), but also for an additional amount of time equal to T_{OXC}. Consequently, the effective service time of a burst follows a general distribution with Laplace transform $B^\star(s)e^{-sT_{\mathrm{OXC}}}$ and mean $1/\mu + T_{\mathrm{OXC}}$.

Based on the above observations, an output port p with W burst wavelengths can be modeled using the $M/G/W/W$ loss system. The traffic intensity $\rho(\mathrm{JET})$ for this system is given by

$$\rho(\mathrm{JET}) = \lambda\left(\frac{1}{\mu} + \overline{T}_{\mathrm{OXC}}\right) \tag{4}$$

and the probability of burst loss at the output port is given by the Erlang-B formula $Erl(\rho(\mathrm{JET}), W)$. Note that, as we discussed above, the $M/G/W/W$ model for JET is approximate since it assumes a Poisson arrival process for bursts (or equivalently, that scheduling decisions are made at the instant a burst arrives, rather than at the time the setup message arrives). It also implies optimal scheduling decisions, when in practice a fast suboptimal algorithm may be used. Nevertheless, numerical results to be presented shortly indicate that this model is quite accurate.

As a final note, the traffic intensity value for JET used in [7] (as well as other studies) does not include the term T_{OXC}, resulting in a lower value than the one in (4). Since these studies ignore the OXC configuration time, their results underestimate the burst loss probability of JET.

4.3 A Model of Horizon

Similar to JET, the length of a wavelength reservation in Horizon is equal to the duration of a burst's transmission *plus* the OXC configuration time T_{OXC}. In order to account for the "no-void-filling" feature of Horizon compared to JET, we let the mean effective service time of bursts be equal to the mean wavelength reservation, $1/\mu + T_{\mathrm{OXC}}$, *plus* a quantity $\Delta \geq 0$. In other words, we use the following value for the traffic intensity of Horizon:

$$\rho(\mathrm{Horizon}) = \lambda\left(\frac{1}{\mu} + \overline{T}_{\mathrm{OXC}} + \Delta\right) \tag{5}$$

We first note that, when the values of the system parameters T_{OXC}, T_{setup}, and $1/\mu$ are such that no void filling is possible in the OBS network (refer to

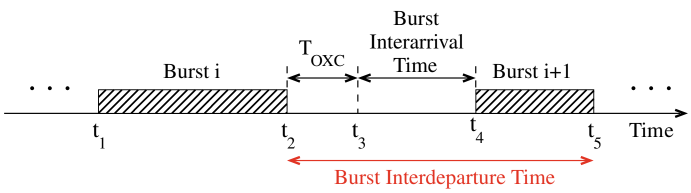

Fig. 9. Departure process of a wavelength in an OBS node with delayed reservation and void filling (JET).

our discussion in Section 5), then obviously, $\Delta = 0$ and Horizon has the same burst drop probability as JET. However, if void filling is possible, then $\Delta > 0$, and the traffic intensity of Horizon is greater than that of JET (refer to Expression (4)), resulting in higher burst drop probability. Using $\Delta > 0$ in (5) implies that the effective service time of bursts is larger than under JET. This increase in the effective service time of bursts has two consequences: first, voids become smaller, and second, the "larger" bursts will not fit within the "smaller" voids. Therefore, the essence of our approximation is to account for the lack of void filling by appropriately increasing the effective service time of bursts, and in turn, the traffic intensity.

In Appendix A, we present an analysis to estimate the value of Δ in Expression (5). Finally, we note that some previous studies, including [7], ignore not only the term T_{OXC} in calculating the traffic intensity of Horizon, but also the additional term Δ we use to account for the lack of void filling. Therefore, these studies clearly underestimate the burst drop probability of Horizon.

4.4 A Model of JIT$^+$

It is possible to obtain approximately the burst drop probability for JIT$^+$ by carrying out an analysis similar to that for Horizon. Specifically, we can obtain the traffic intensity value as in (5), but replace Δ with a new quantity $\Delta' > \Delta$. The new larger value Δ' would account for both the lack of void filling and the limit of at most two delayed reservations per wavelength. However, we have found that estimating the value of Δ' using analytical techniques is a complicated and difficult task. Therefore, we have decided to use simulation to obtain the burst drop probability of JIT$^+$.

Discussion

If we ignore the differences in the setup message processing time $T_{\mathrm{setup}(X)}$ among the three reservation schemes X, then, in general, JET will result in the lowest burst drop probability, followed by Horizon, JIT$^+$, and JIT. In practice, however, the relative performance of the four schemes depends on the actual values of certain system parameters. Let $X \equiv Y$ denote that reservation scheme X is *equivalent* to scheme Y (in the sense that both result in *the same* burst drop probability), and $X \approx Y$ denote that schemes X and Y result in *approximately the same* burst drop probability. Then, we can make the following observations.

- $T_{\mathrm{OXC}} > kT_{\mathrm{setup}} \Rightarrow$ JET $\equiv$ Horizon $\equiv$ JIT$^+$ Referring to (1), if T_{OXC} is larger than the sum of setup message processing times, then no void filling may take place. This is because two OXC configuration operations are needed for a burst with a later setup message to fill a void created by a burst with an earlier setup message: one operation to switch the former burst, and one to switch the latter. The total time required for these operations is $2T_{\mathrm{OXC}}$, while the void is at most equal to $T_{\mathrm{offset}} = T_{\mathrm{OXC}} + kT_{\mathrm{setup}} \leq 2T_{\mathrm{OXC}}$. Therefore, JET reduces to Horizon in this case. Interestingly, if the above condition is true, a wavelength cannot be reserved for more than two bursts at any given time. To see this, refer to Fig. 7. In order to have a third reservation under Horizon, the setup message of the third burst (burst $i+2$ in the figure) must arrive before the end of the first burst (burst i in the figure) at time t_4. However, it is clear from the figure that the interval from time t_4 to the time horizon t_8 is at least equal to $2T_{\mathrm{OXC}}$, i.e., it is greater than T_{offset}. As a result, any burst whose setup message arrives before time t_4 would be dropped by the OBS node. Consequently, Horizon (and JET) also reduces to JIT$^+$. This case is of practical interest because of the state-of-the-art in OXC technologies in the foreseeable future.
- Minimum burst length $+T_{\mathrm{OXC}} > kT_{\mathrm{setup}} \Rightarrow$ JET $\equiv$ Horizon $\equiv$ JIT$^+$ For similar reasons, if the minimum burst length plus the OXC configuration time T_{OXC} is larger than the sum of processing times, then (1) no void filling is possible, and (2) at most two bursts can reserve a wavelength at any given time, hence both JET and Horizon reduce to JIT$^+$.
- $T_{\mathrm{offset}} = \{$constant $\Rightarrow$ JET $\equiv$ Horizon

If the offset value is constant (rather than equal to the minimum value in (1)), then no void filling is possible therefore JET reduces to Horizon. Note that a constant offset value may be of practical importance. For example, rather than estimating the number of hops to the destination in order to compute the minimum offset value according to (1), it may be desirable to set the offset to a large value that can accommodate any source-destination pair; this is similar to setting the TTL of an IP packet to a high value rather than one based on a given source-destination pair. Furthermore, if alternate routing algorithms are used to reduce the burst loss probability, as has been suggested in the literature, then the number of hops in the actual path may not be easy to estimate; a large constant offset value might then be appropriate.

- $(1/\mu \gg T_{\mathrm{OXC}} \text{ and } 1/\mu \gg T_{\mathrm{setup}}) \Rightarrow \mathrm{JET} \approx \mathrm{Horizon} \approx \mathrm{JIT}^{+} \approx \mathrm{JIT}$

If the mean burst size $1/\mu$ is large relative to the values of T_{OXC} and T_{setup}, then from (1), it is also large with respect to T_{offset}. As a result, there are few opportunities for void filling or delayed reservations, and the performance of all four schemes will be very similar. We can reach the same conclusion by observing that, in this case, the traffic intensity value of JIT, JET, and Horizon (see (3), (4), and (5)) is dominated by $1/\mu$, resulting in similar burst drop probabilities for the three schemes, as well as for JIT^{+} whose performance lies between that of JIT and Horizon. Note that T_{OXC} and T_{setup} represent the overheads associated with switching bursts in the network. Therefore, it is reasonable to assume that, whatever the actual values of these parameters, the mean burst length must be significantly larger, otherwise the network will waste a large fraction of its resources on overhead operations rather than on transmitting bursts, resulting in low throughput or high burst drop probability regardless of the reservation scheme used.

- As a burst travels along its path, its offset value decreases by an amount equal to T_{setup} for each OBS node visited. As a result, inside the network, the offset value becomes dominated by T_{OXC} (refer to (1)), and all four reservation schemes will have similar performance. Consequently, the JET or Horizon schemes may offer the highest benefit at edge nodes, rather than inside the network.

5 Numerical Results

In this section we compare the JIT, JIT^{+}, JET, and Horizon schemes in terms of burst loss probability. In our comparison we consider both a single OBS node in isolation (see Section 5.1) and a path of OBS networks with cross-traffic (Section 5.2). For the single OBS node, we use the Erlang-B formula (2) with the appropriate traffic intensity to obtain the burst loss probability. Since this formula is exact only for JIT, we also use simulation for the other three reservations schemes to estimate the burst loss probability. For the path OBS network, we use simulation for all four reservation schemes. In obtaining the simulation results, we have estimated 95% confidence intervals using the method of batch means. The number of batches is 30, with each batch run lasting until at least 120,000 bursts are transmitted by each OBS node. However, we have found that the confidence intervals are very narrow. Therefore, to improve readability, we do not plot the confidence intervals in the figures we present in this section.

In our comparisons, we use six sets of values for the various system parameters, as shown in

Table 1. Values of the system parameters for the various traffic scenarios used in the performance comparison.

State of Technology	Scenario	$1/\mu$	T_{OXC}	$T_{\mathrm{setup}}(\mathrm{JIT}) = T_{\mathrm{setup}}(\mathrm{JIT}^{+})$	$T_{\mathrm{setup}}(\mathrm{Horizon}) = 2T_{\mathrm{setup}}(\mathrm{JIT})$	$T_{\mathrm{setup}}(\mathrm{JET}) = 4T_{\mathrm{setup}}(\mathrm{JIT})$
Current	1	50 ms	10 ms	12.5 μs	25 μs	50 μs
	2	10 ms	10 ms	12.5 μs	25 μs	50 μs
Near	3	100 μs	20 μs	1 μs	2 μs	4 μs
Future	4	20 μs	20 μs	1 μs	2 μs	4 μs
Distant	5	2.5 μs	500 ns	50 ns	100 ns	200 ns
Future	6	500 ns	500 ns	50 ns	100 ns	200 ns

Table 1. Scenarios 1 and 2 in the table correspond to T_{OXC} and T_{setup}(JIT) values that reflect currently available technology. Specifically, we let $T_{OXC} = 10$ ms, a value that represents the configuration time of existing MEMS switches [23], and $T_{setup}(\text{JIT}) = T_{setup}(\text{JIT}^+) = 12.5\,\mu s$, a value that corresponds to the processing time of JIT signaling messages in our JITPAC controllers [21]. To the best of our knowledge, the JET and Horizon schemes have not been implemented in hardware, therefore we do not have actual values for $T_{setup(JET)}$ or T_{setup}(Horizon). Therefore, we estimate their values to be four and two times, respectively, the value of T_{setup}(JIT), and we use these relative values for all scenarios we consider. In particular, $T_{setup}(\text{JET}) = 50\,\mu s$, $T_{setup}(\text{Horizon}) = 25\,\mu s$, for the current scenario. We emphasize that while these values are only best guess estimates, we have found that the relative performance of the four schemes is not significantly affected as long as these values are a small multiple of T_{setup}(JIT).

In Scenarios 1 and 2, we use the same values of T_{OXC} and $T_{setup(X)}$ for all four reservation schemes X. The main difference between the two scenarios is that in Scenario 1 we let the mean burst size $1/\mu = 5T_{OXC} = 50\,ms$, while in Scenario 2 we let $1/\mu = T_{OXC} = 10\ ms$. As we noted in the previous section, the smaller the value of the mean burst size relative to T_{OXC} or T_{setup}, the larger the fraction of time the OBS nodes spend on overhead operations, and the lower the throughput; this result is borne out in the results we present in this section.

Scenarios 3 and 4 in Table 1 correspond to projections regarding the state of OXC and hardware processing technology in the near future (e.g., in 3–5 years). Specifically, we let $T_{OXC} = 20\ \mu s$ (an improvement of three orders of magnitude over the previous scenario) and $T_{setup}(\text{JIT}) = T_{setup}(\text{JIT}^+) = 1\,\mu s$ (an improvement of one order of magnitude). These projections assume that the less mature OXC technology will improve faster than the more mature hardware processing technology. The values of Tsetup(JET) and Tsetup(Horizon) relative to Tsetup(JIT) are the same as above. Also, the difference between Scenario 3 and Scenario 4 is that the mean burst size takes values equal to 5TOXC and TOXC, respectively.

Scenarios 5 and 6 represent projections regarding the state of the technology in the more distant future. In this scenario, we assume that OXC configuration times will improve to 500 ns, and setup processing times for JIT and JIT$^+$ will decrease to 50 ns. The relative values of T_{setup}(JET) and T_{setup}(Horizon), as well as the values of the mean burst size $1/\mu$ are the same as in the previous pairs of scenarios.

In our study, we also assume that the number of hops in the path of a burst is uniformly distributed between 1 and 10, and we calculate the offset using (1). The arrival rate λ of setup messages is such that $\lambda/\mu = 32$ for *all* scenarios. Finally, in the simulation, we used the latest available unused channel (LAUC) algorithm [11,13] in JET and Horizon to select an available wavelength for an arriving burst; for JIT and JIT$^+$, any of the available wavelengths was selected with equal probability to transmit a new burst.

5.1 A Single OBS Node

The six Figures 10–15 plot the burst drop probability of JET, Horizon, JIT$^+$, and JIT, as the number W of wavelengths varies from 8 to 64, for the six scenarios listed in Table 1, respectively. Recall that Scenarios i and $i+1$, $i = 1,3,5$, have the same values for the system parameters, but use different mean burst lengths: for Scenario i we have that $1/\mu = 5T_{OXC}$, while for Scenario $i+1$ we have used $1/\mu = T_{OXC}$. As a result, Scenario $i+1$ presents more opportunities for delayed reservations and void filling than Scenario i, $i = 1,3,5$, which JET and Horizon can take advantage of. However, these opportunities come at the expense of higher switching overheads relative to the mean burst size; hence, we expect the overall burst drop probability to be higher in Scenario $i+1$ than in Scenario i.

Because of the high value of the arrival rate λ relative to the mean burst size ($\lambda/\mu = 32$), the burst drop probability is high for up to $W = 32$ wavelengths. Under Scenarios 1, 3, and 5 (Figures 10, 12, and 14), the burst drop probability decreases dramatically for $W = 64$, and becomes zero for $W = 128$ (not shown in the figures). On the other hand, the burst drop probability remains quite high (around 10%) under Scenarios 2, 4, and 6 (Figures 11, 13, and 15). This behavior is due to the high burst switching and

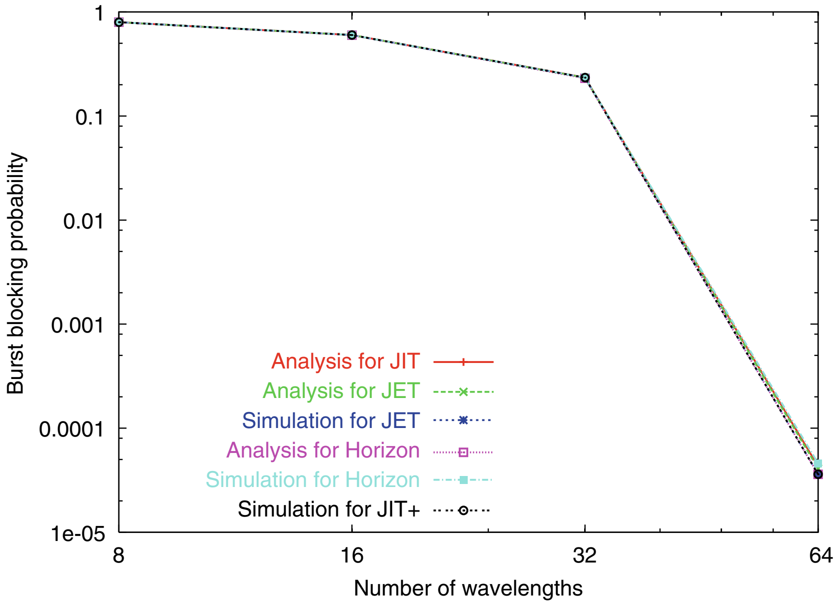

Fig. 10. Single node performance comparison, Scenario 1 (current technology, $1/\mu = 5\ T_{\mathrm{OXC}}$).

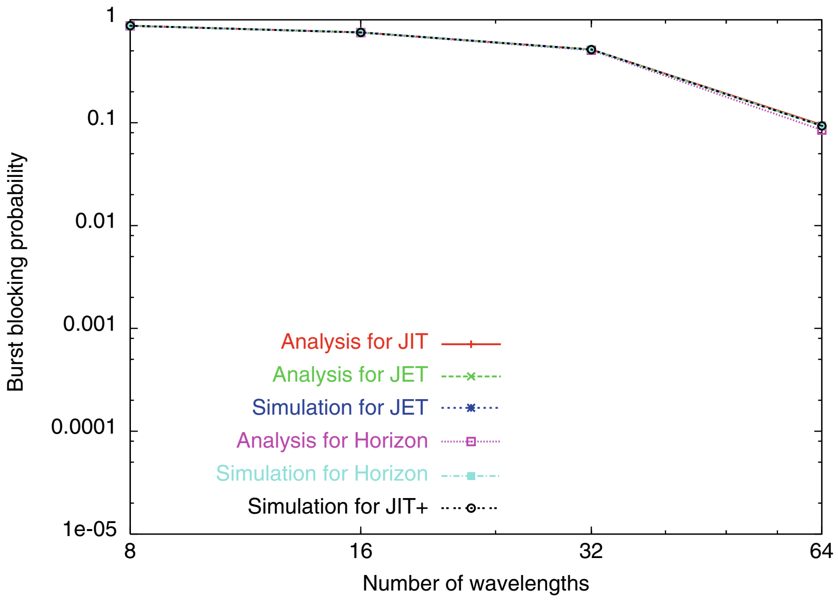

Fig. 11. Single node performance comparison, Scenario 2 (current technology, $1/\mu = T_{\mathrm{OXC}}$).

setup message processing overheads when the mean burst size is small relative to T_{OXC} and T_{setup}. Consequently, these results indicate that, regardless of the values of T_{OXC} and T_{setup}, the mean burst size must be significantly larger otherwise the network will suffer either high burst drop probability or low utilization (if the offered load is reduced to yield an acceptable burst drop probability).

In all six figures, we observe the good match between analytical and simulation results for JET and Horizon, across all sets of values for the sys-

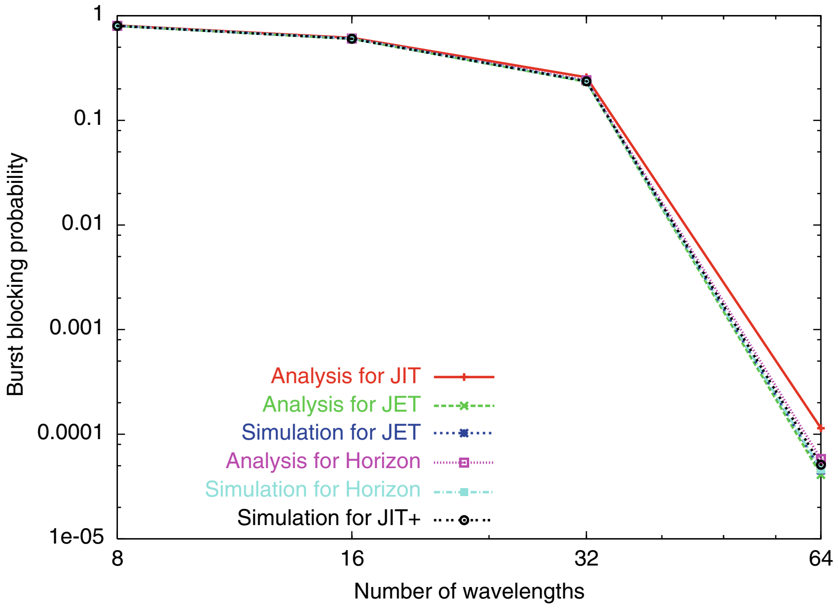

Fig. 12. Single node performance comparison, Scenario 3 (near future technology, $1/\mu = 5\ T_{OXC}$).

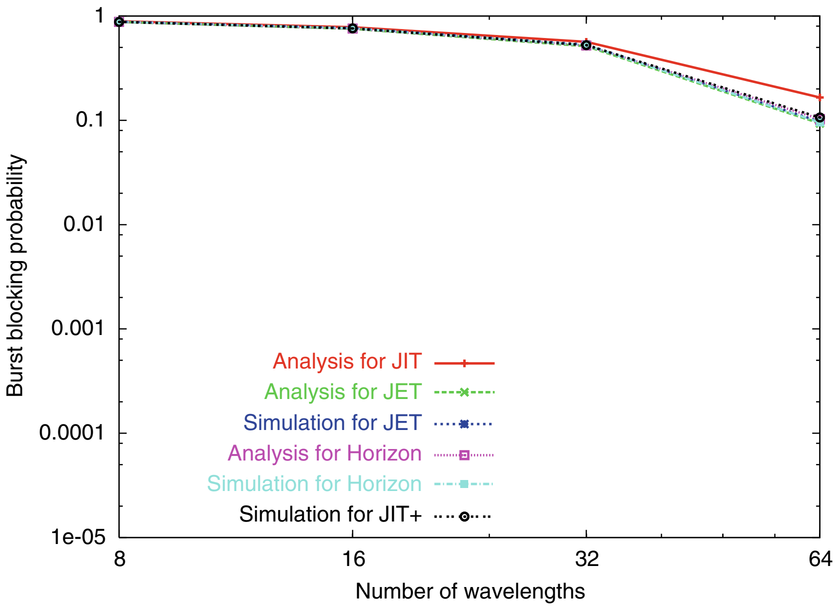

Fig. 13. Single node performance comparison, Scenario 4 (near future technology, $1/\mu = T_{OXC}$)

tem parameters as well as across the various values of W. More importantly, we observe that the burst probability of the JET, Horizon, and JIT$^+$ reservations schemes is very similar, and in most cases identical. Under the odd-numbered scenarios, JIT has similar performance with the other three schemes, except when the number of wavelengths increases beyond 32. Under the even-numbered scenarios, on the other hand, the performance of JIT, which does not allow any delayed reservations, lags that of the other three schemes, as expected. However, the fact that the burst drop

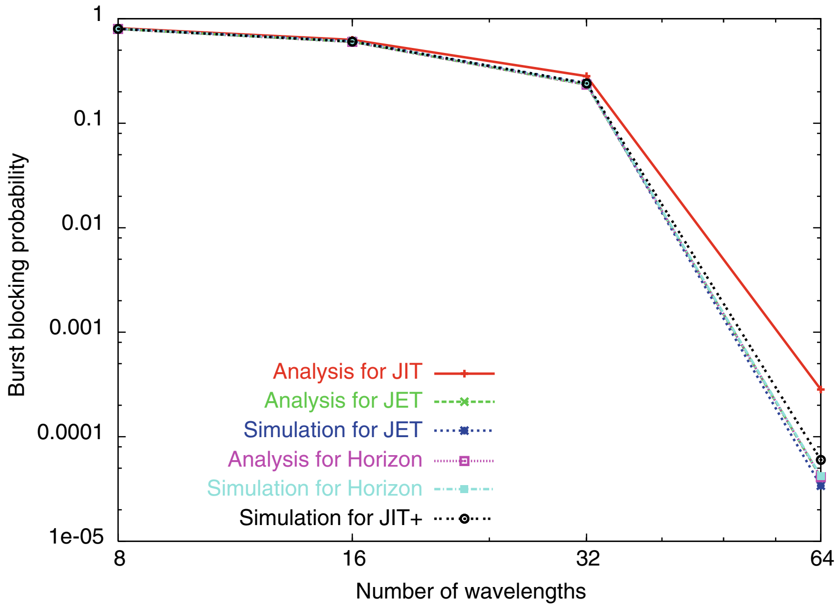

Fig. 14. Single node performance comparison, Scenario 5 (distant futuretechnology, $1/\mu = 5T_{OXC}$).

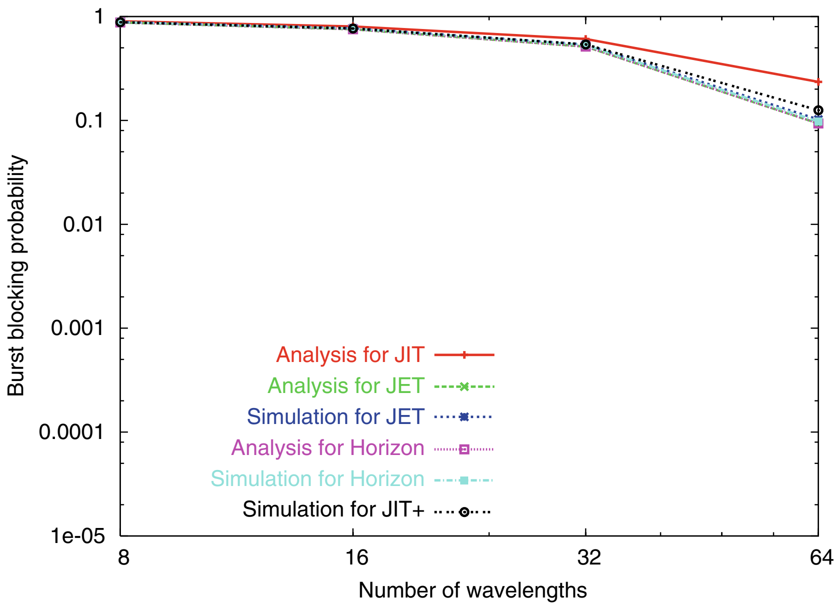

Fig. 15. Single node performance comparison, Scenario 6 (near future technology, $1/\mu = T_{OXC}$).

probability of JIT$^+$ tracks that of JET and Horizon well, indicates that it is possible to achieve good performance with a scheme of modest complexity, effectively simplifying the design and operation of OBS nodes.

Overall, our results show that, for T_{OXC} and T_{setup} values corresponding to the state of the technology today and in the foreseeable future, and for burst lengths that are not dominated by the switching and processing overheads, there is little opportunity for performing void filling or delayed reservations of more than two bursts on a given wavelength. As a result, the JIT$^+$ scheme performs similarly to JET and Horizon, making it a good choice for emerging OBS testbeds. On the other hand, we have found that JET and Horizon perform better than JIT$^+$ when the mean burst size is at least an order of magnitude smaller than T_{setup} and T_{OXC}, in which case there is ample opportunity for void filling and/or delayed reservations of multiple bursts. However, as we discussed in the previous section, and as Figures 11, 13, and 15 illustrate, it is highly unlikely that OBS networks will be designed to operate under such a scenario, since the high switching and processing overhead would result in very high burst drop probabilities and low throughput.

5.2 A path of an OBS Network

In order to compare the performance of the four wavelength reservation schemes along a path of a network with cross traffic, we now consider the linear OBS network shown in Fig. 16. The network consists of k OBS nodes connected in a unidirectional linear topology in which traffic flows from left to right only. Each OBS node (except S_k) serves exactly N users that can transmit bursts. The traffic pattern in the linear network is as follows. The N users of node S_1 generate bursts whose destination is one of the nodes S_2 to S_k. The destination of a burst is uniformly distributed among S_2 and S_k, thus the number of hops in the path of a burst is also uniformly distributed between 1 and $k-1$. We will refer to the traffic generated from node S_1 as *through traffic*. The N users of node $S_i, i = 2, \cdots, k-1$, generate bursts which travel along the link from node S_i to node S_{i+1} and then leave the network, as illustrated in Fig. 13. In calculating the offset for these bursts, we also assume that the number of hops in their paths is uniformly distributed between 1 and $k-1$. The traffic from node S_i, $i = 2, \cdots, k-1$, to node S_{i+1} will be referred to as *cross traffic*.

In our experiments, we simulated a path network with $k=11$ nodes. Our simulation model accounts for the transmission of both setup messages and bursts, as well as for the processing times and OXC configuration times at each OBS node. We used the same scenarios and parameter values listed in Table 1, and we let the arrival rate λ of setup messages be such that $\lambda/\mu = 32$ for *all* scenarios.

Figures 17–22 plot the burst drop probability of the *through traffic* (i.e., traffic from node S_1 to all other nodes), in the path of Fig. 13; each figure corresponds to one of the scenarios listed in Table 1. Note that the burst drop probability in each of these figures is much higher than the corresponding figure of the previous subsection, since the through bursts have to be switched by up to $k=10$ nodes, at each node competing with cross-traffic bursts for switching resources. We also note again that, under scenarios in which the mean burst length is small relative to T_{OXC}, the burst drop probability is higher than scenarios in which the mean burst length is large, confirming our previous observations regarding the desirable region of network operation. (For odd-numbered scenarios, the burst drop probability for $W=128$ wavelengths is zero.)

Regarding the relative performance of the four wavelength reservation schemes, we again observe that JET, Horizon, and JIT$^+$ have similar behavior across the different scenarios and number of wavelengths. We also observe, however, that when the number of wavelengths is not too large, JIT results in lower through burst drop probability than the other three schemes. To explain this surprising result we carefully studied the simulation results, and we found that the higher drop probability of JET, Horizon, and JIT+ is mainly due to the loss of

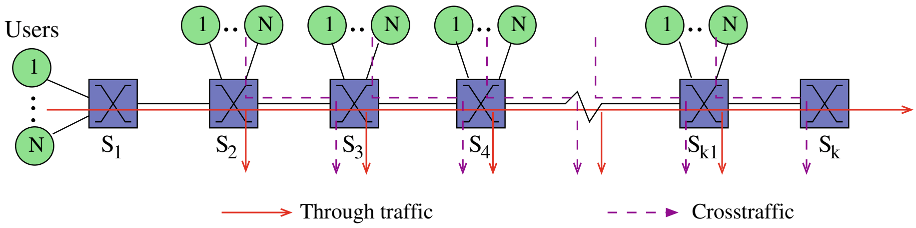

Fig. 16. The linear OBS network.

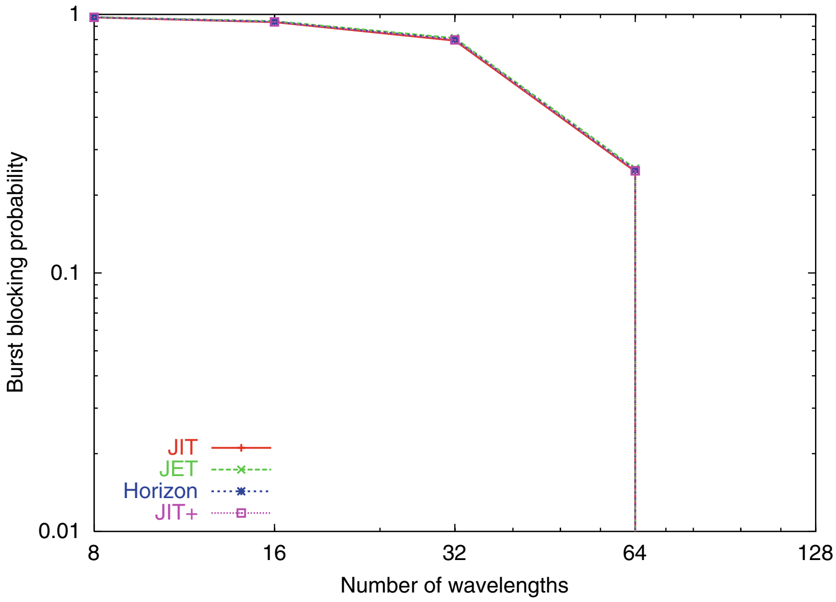

Fig. 17. Path performance comparison, Scenario 1 (current technology, $1/\mu = 5T_{\mathrm{OXC}}$).

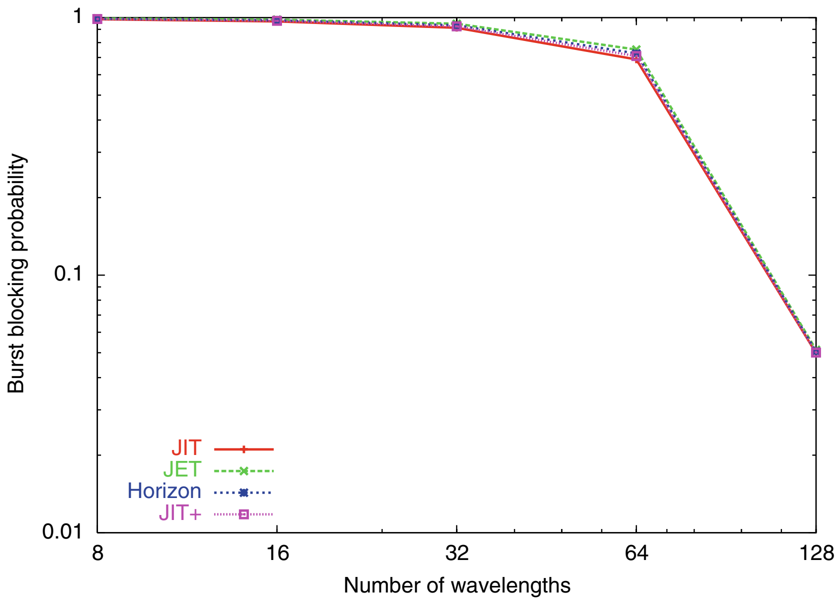

Fig. 18. Path performance comparison, Scenario 2 (current technology, $1/\mu = T_{\mathrm{OXC}}$).

large numbers of through bursts whose destinations are close to the source node S_1. Note that through bursts that travel only a few hops have a short offset. It is well-known that, for reservation schemes, such as JET, Horizon, and JIT, that allow delayed reservations and/or void filling, a shorter offset for through bursts results in lower priority with respect to competing cross-traffic bursts, hence higher drop probability [10]. On the other hand, in JIT, it is the arrival time of the burst, not its offset length, that determines whether the burst will be accepted or not. Consequently, the number of dropped through bursts that have to travel only a few hops is

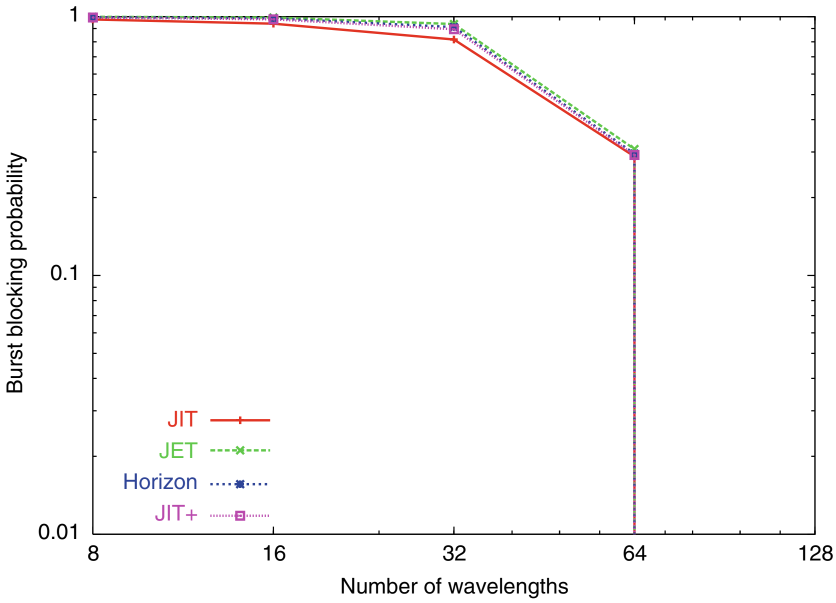

Fig. 19. Path performance comparison, Scenario 3 (near future technology, $1/\mu = 5T_{\mathrm{OXC}}$).

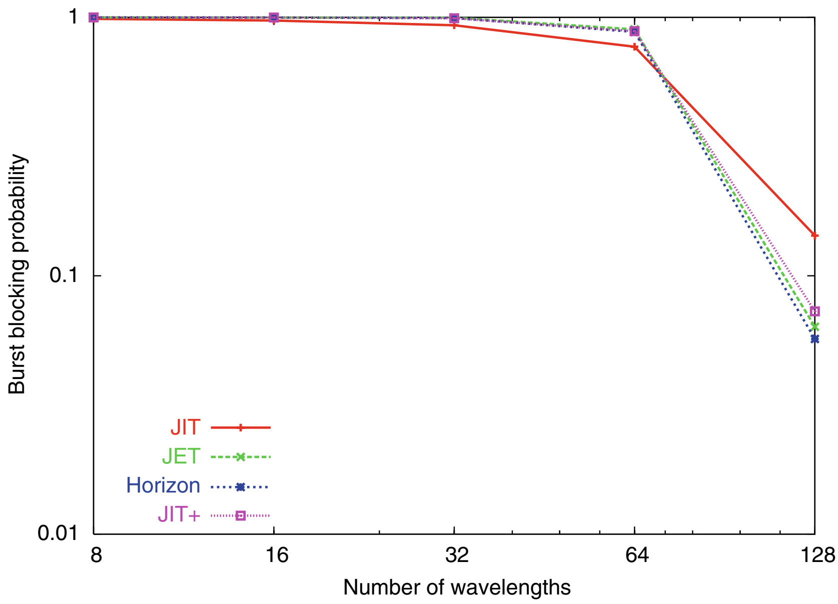

Fig. 20. Path performance comparison, Scenario 4 (near future technology, $1/\mu = T_{\mathrm{OXC}}$).

significantly smaller than the other three schemes, resulting in smaller overall burst drop probability when the number of wavelengths is not very large; note that this counter-intuitive behavior of JET and Horizon has not been observed before. When the number of wavelengths increases sufficiently, however, the other three schemes exhibit better performance than JIT on a per-node basis and their through burst drop probability is lower than that of JIT.

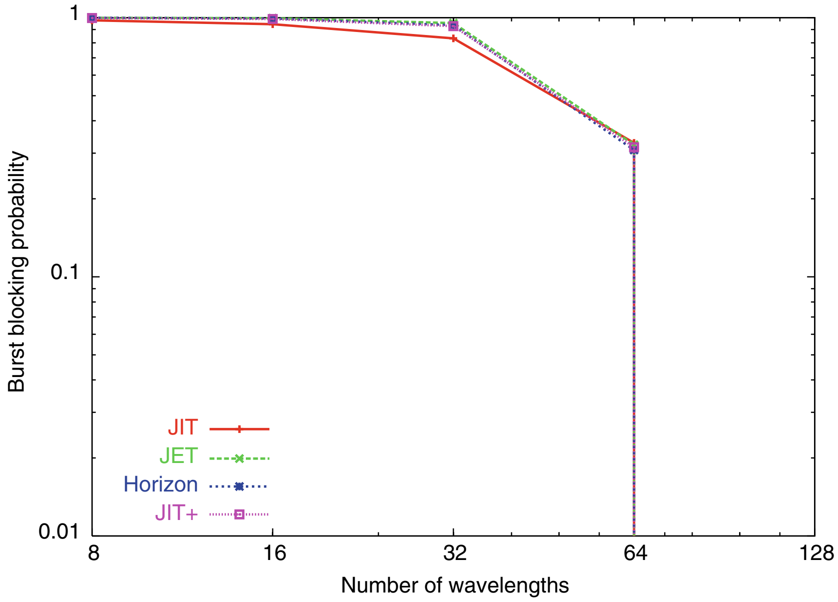

Fig. 21. Path performance comparison, Scenario 5 (distant future technology, $1/\mu = 5T_{OXC}$).

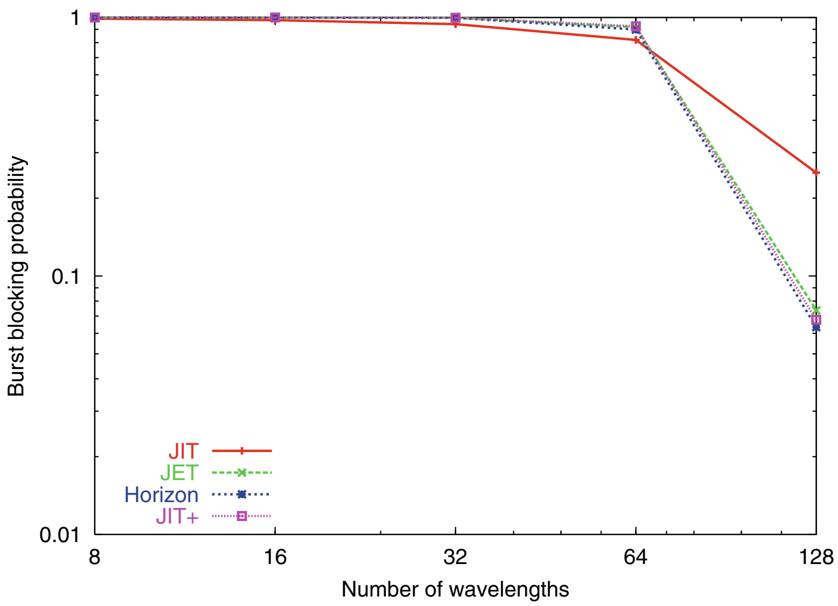

Fig. 22. Path performance comparison, Scenario 6 (distant future technology, $1/\mu = T_{OXC}$).

6. Concluding Remarks

We have presented a detailed analysis of the JIT, JET, and Horizon wavelength reservation schemes for OBS networks, and we have introduced a new reservation scheme, JIT$^+$. We have also presented numerical results to compare the performance of the four schemes in terms of burst drop probability under a range of network scenarios. Our work accounts for the switching and processing overheads associated with bursts as they travel across the network, and it provides new insight into the relative capabilities of the various schemes. Our findings indicate that the simpler JIT and JIT$^+$ reservation schemes appear to be a good choice for the foreseeable future. Jointly with MCNC-RDI, we have developed a complete specification of a JIT signaling protocol, and have implemented it in a proof-of-concept OBS testbed. We are currently working on extending the specification to include the new JIT$^+$ reservation scheme.

Appendix A: Estimation of the Parameter Δ for Horizon

We now consider the problem of estimating the value of parameter Δ in the Expression (5) for the traffic intensity of Horizon. Recall that Δ represents the increase in the effective service time of bursts under Horizon over that under JET, to prevent any void filling from taking place. In the following analysis, we consider a single wavelength $w, w = 1, \cdots, W$, of the output port in isolation. Assuming that the burst scheduling algorithm is not biased to favor some wavelengths over the others, then, in the long run, we can assume that the arrival rate of bursts to each wavelength is equal to λ/W. Reasoning about the departure process of Horizon becomes much easier when there is a single output wavelength, and, comparing to simulation results, we have found that the results of considering each wavelength in isolation are reasonably accurate.

Let us refer to Fig. 5 which shows the burst departure process on a single wavelength. We note that, because of the additional burst dropping (compared to JET) due to the lack of void filling, the mean length of the interval $t_6 - t_5$ is greater than the mean burst interarrival time W/λ. The essence of our approximation is to increase the effective service time of bursts by an amount equal to the difference between the mean length of this interval and the mean burst interarrival time.

We now show how to find the distribution of the length u of the interval of time between t_5 and t_6 in Fig. 5. This interval corresponds to the time until the next burst arrival, since any burst arriving after time t_5 is accepted. We let $Prob^{\text{noburst}}(u)$ denote the probability that no burst arrives in an interval of length u; note that we assume that this probability depends only on the length of the interval, not its start time.

Let us define the *holding time* of a burst as the sum of three quantities: (1) the burst offset, (2) the burst length, and (3) the OXC configuration time T_{OXC}. From Fig. 5, we observe that burst $i+1$ is the first burst whose setup message arrives after the arrival of burst i's setup message and whose first bit arrives after the end of the holding time of burst i (i.e. t_5). In other words, all the bursts with setup messages arriving between t_1 and t_2 must have completed their offset before t_5. Therefore, to analyze the interval between the end of the holding time of burst i and the arrival of burst $i+1$, we only need to consider those bursts whose setup messages arrive between t_1 and t_2. Thus we can initiate a new busy period at time t_1, so t_1 is time 0 in this new busy period.

Let s denote the holding time of a burst, which is distributed according to CDF $H(s)$; the Laplace transform of this CDF can be easily obtained from the definition above. Let also $t = t_2 - t_1$ denote the interval between the arrival times of the setup messages of bursts i and burst $i+1$.

From [25], we know that for a Poisson arrival process, with a certain number of customers arriving within a given period, the arrival times of these customers are uniformly distributed in that period. Thus, the probability that a customer arriving in $(0,u)$ is still in the system at time u' is $\frac{1}{u}\int_0^u [1 - G(u' - x)]dx$, where $G(z)$ is the CDF of the offset length. Then, the probability that the k bursts whose setup message arrives in the period $(0,t)$ would have their first bit arrive before time s is $\left[\frac{1}{t}\int_0^t G(s-x)dx\right]^k$.

The sum of $k+1$ exponentially distributed intervals follows a $(k+1)$-stage Erlang distribution, so the PDF of t is $\frac{\lambda/W(\lambda t/W)^k e^{-\lambda t/W}}{k!}$. Therefore, the probability that all the bursts whose setup messages arrive in the period $(0,t)$ would have their first bit arrive before time s is

$$\sum_{k=0}^{\infty} \frac{\lambda}{W} e^{-\lambda t/W} \frac{(\lambda t/W)^k}{k!} \left[\frac{1}{t}\int_0^t G(s-x)dx\right]^k = \frac{\lambda}{W} \mathrm{e}^{-\lambda/W\left[t - \int_0^t G(s-x)dx\right]} \tag{6}$$

Now, the probability that burst $i+1$ (whose setup message arrives at time t) has an offset greater than

$s+u$ is $1-G(s+u-t)$, and the probability that no burst arrives during the interval $(s,s+u)$ is

$$Prob^{\text{noburst}}(u) = \int_{s=0}^{\infty}\int_{t=0}^{\infty}\frac{\lambda}{W}\mathrm{e}^{-\lambda\left[t-\int_0^t G(s-x)dx\right]/W}[1-G(s+u-t)]\mathrm{d}t\,\mathrm{d}H(s) \tag{7}$$

The CDF of u is $P(u)=1-Prob^{\text{noburst}(u)}$, and we obtain the expected value of u as

$$\overline{u} = \int_0^{\infty} udP(u) = \int_0^{\infty}(1-P(u))\mathrm{d}u \tag{8}$$

Given the CDF $G(z)$ and $H(s)$, it is possible to compute $\overline{u}$ numerically. We then let $\Delta = \frac{\overline{u}}{W} - \frac{1}{\lambda}$ in the expression (5) for the traffic intensity of Horizon.

References

[1] C. Qiao, M. Yoo, Optical burst switching (OBS)-A new paradigm for an optical Internet, Journal of High Speed Networks, vol. 8, no. 1, (January 1999), pp. 69–84.

[2] J. S. Turner, Terabit burst switching, Journal of High Speed Networks, vol. 8, no. 1, (January 1999), pp. 3–16.

[3] J. Y. Wei, R. I. McFarland, Just-in-time signaling for WDM optical burst switching networks, IEEE/OSA Journal of Lightwave Technology, vol. 18, no. 12, (December 2000), pp. 2019–2037.

[4] I. Baldine, G. N. Rouskas, H. G. Perros, D. Stevenson, Jumpstart: A just-in-time signaling architecture for WDM burst-switched networks, IEEE Communications Magazine, vol. 40, no. 2, (2002), pp. 82–89.

[5] M. Duser, P. Bayvel, Analysis of a dynamically wavelength-routed, optical burst switched network architecture, IEEE/OSA Journal of Lightwave Technology, vol. 20, no. 4, (April 2002), pp. 574–585.

[6] J. Y. Wei, J. L. Pastor, R. S. Ramamurthy, Y. Tsai, Just-in-time optical burst switching for multiwavelength networks. In IFIP TC6 WG6.2 Fifth International Conference on Broadband Communications, (Kluwer Academic Publishers, November 1999). pp. 339–352.

[7] K. Dolzer, C. Gauger, J. Späth, Evaluation of reservation mechanisms for optical burst switching, AEÜ International Journal of Electronics and Communications, vol. 55, no. 1, (January 2001), pp. 18–26.

[8] K. Dolzer, C. Gauger, On burst assembly in optical burst switching networks – a performance evaluation of Just-Enough-Time, In Proceedings of the 17th International Teletraffic Congress, (Salvador, Brazil, September 2001), pp. 149–161.

[9] S. Verma, H. Chaskar, R. Ravikanth, Optical burst switching: a viable solution for terabit IP backbone, IEEE Network, vol. 14, no. 6, (November/December 2000), pp. 48–53.

[10] M. Yoo, C. Qiao, S. Dixit, QoS performance of optical burst switching in IP-over-WDM networks, Journal on Selected Areas in Communications, vol. 18, no. 10, (October 2000), pp. 2062–2071.

[11] Y. Xiong, M. Vandenhoute, H. C. Cankaya, Control architecture in optical burst-switched WDM networks, IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, (October 2000), pp. 1838–1851.

[12] F. Callegati, H. C. Cankaya, Y. Xiong, M. Vandenhoute, Design issues of optical IP routers for internet backbone applications, IEEE Communications Magazine, vol. 37, no. 12, (1999), pp. 124–128.

[13] J. Xu, C. Qiao, J. Li, G. Xu, Efficient channel scheduling algorithms in optical burst switched networks, In Proceedings of IEEE INFOCOM, (San Francisco, USA, April 2003), vol. 3, pp. 2268–2278.

[14] M. Yang, S. Q. Zheng, D. Verchere, A QoS supporting scheduling algorithm for optical burst switching DWDM networks. In Global Telecommunications Conference, (San Antonio, Texas, November 2001) vol. 1, pp. 86–91.

[15] C. Gauger, Dimensioning of FDL buffers for optical burst switching nodes, In Proceedings of the 5th IFIP Optical Network Design and Modeling Conference (ONDM 2002), (Torino, Italy, February 2002).

[16] A. Ge, F. Callegati, L. S. Tamil, On optical burst switching and self-similar traffic, IEEE Communications Letters, vol. 4, no. 3, (March 2000), pp. 98–100.

[17] X. Yu, Y. Chen, C. Qiao, A study of traffic statistics of assembled burst traffic in optical burst switched networks, In Proceedings of Opticomm, (Denver, Colorado, August 2001), pp. 149–159.

[18] http://jumpstart.anr.mcnc.org.

[19] http://www.ic-arda.org.

[20] I. Baldine, G. N. Rouskas, H. G. Perros, D. Stevenson, Signaling support for multicast and QoS within the JumpStart WDM burst switching architecture, Optical Networks, vol. 4, no. 6, (November/December 2003), pp. 68–80.

[21] I. Baldine, M. Cassada, A. Bragg, G. Karmous-Edwards, D. Stevenson, Just-in-time optical burst switching implementation in the ATDnet all-optical networking testbed, In Proceedings of Globecom 2003, (San Francisco, USA, December 2003), vol. 5, pp. 2777–2781.

[22] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, D. Stevenson, Jumpstart just-in-time signaling protocol: A formal description using extended finite state machines, Optical Engineering, vol. 42, no. 2, (February 2003), pp. 568–585.

[23] P. B. Chu, S. S. Lee, S. Park, MEMS: The path to large optical crossconnects, IEEE Communications Magazine, vol. 40, no. 3, (March 2002), pp. 80–87.

[24] M. Yoo, C. Qiao, Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks, In IEEE/

LEOS Technol. Global Information Infrastructure, (Montreal, Canada, August 1997), pp. 26–27.
[25] L. Kleinrock, Queueing Systems The Volume 1: Theory (John Wiley & Sons, New York, 1975).

Jing Teng received a B.S. in Computer Science from Wuhan University, Wuhan, China, in 1997, and a M.S. in Computer Science from the Chinese Academy of Sciences, Beijing, China, in 2000. In 2000, he joined the Department of Computer Science at NC State University, Raleigh, NC, U.S., to pursue a Ph.D. degree. His research interests include network protocols, optical networks, and performance evaluation.

George N. Rouskas is a Professor of Computer Science at North Carolina State University. He received the Diploma in Computer Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. During the 2000–2001 academic year he spent a sabbatical term at Vitesse Semiconductor, Morrisville, NC, and in May 2000 and December 2002 he was an Invited Professor at the University of Evry, France. His research interests include network architectures and protocols, optical networks, multicast communication, and performance evaluation. Dr. Rouskas received the 2004 ALCOA Foundation Engineering Research Achievement Award, and the 2003 NCSU Alumni Outstanding Research Award. He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award, a co-author of a paper that received the Best Paper Award at the 1998 SPIE conference on All-Optical Networking, and the recipient of the 1994 Graduate Research Assistant Award from the College of Computing, Georgia Tech. Dr. Rouskas is especially proud of his teaching awards, including his induction in the NCSU Academy of Outstanding Teachers in 2004, and the Outstanding New Teacher Award he received from the Department of Computer Science in 1995. Dr. Rouskas has been on the editorial boards of the *IEEE/ACM Transactions on Networking*, *Computer Networks*, and *Optical Networks*, and he was a co-guest editor for the *IEEE Journal on Selected Areas in Communications*, Special Issue on Protocols and Architectures for Next Generation Optical WDM Networks, published in October, 2000. He was Technical Program co-chair of the Networking 2004 conference, and Program Chair of the IEEE LANMAN 2004 workshop. He is a senior member of the IEEE, and a member of the ACM and of the Technical Chamber of Greece.