ABSTRACT

Teng, Jing A Study of Optical Burst Switched Networks with the Jumpstart Just In Time Signaling Protocol. (Under the direction of Professor George N. Rouskas).

This thesis studies the optical burst switched (OBS) networks. It consists of three parts. In the first part, we present a detailed analysis of three major wavelength reservation schemes for OBS networks: JIT, JET, and Horizon. The contributions include: **(i)** analytical models of JET and Horizon (on a single OBS node) that are more accurate than previously published ones and valid for general burst length and offset length distributions; **(ii)** determination of the regions of parameter values, such as burst offset length, optical switching and hardware processing overheads, so that 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, and its performance tracks that of Horizon and JET. We compare the performance (in terms of burst drop probability) 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 technologies in optical switch and electronic hardware, the simplicity of JIT and JIT^+ seems to outweigh any performance benefits of Horizon and JET.

In the second part of this thesis, we present the results of a simulation analysis of OBS networks employing the `Jumpstart` JIT signaling protocol. We study the performance of various network topologies in terms of burst drop probability and investigate the effects of several system parameters, including message processing time, OXC configuration delay, user-to-switch propagation delay, and switch-to-switch propagation delay. We also investigate the effect of wavelength converters.

In the third part, we develop a suite of adaptive and non-adaptive wavelength assignment policies for OBS networks. We also apply traffic engineering techniques to reduce wavelength contention through traffic isolation. Our performance study indicates that, in the absence of full conversion capabilities, intelligent choices in assigning wavelengths to bursts at the source can have a profound effect on the burst drop probability in an OBS network.

A Study of Optical Burst Switched Networks with the Jumpstart Just In Time Signaling Protocol

by

Jing Teng

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Department of Computer Science

Raleigh

2004

Approved By:

Dr. Khaled Harfoush	Dr. Carla D. Savage
Dr. George N. Rouskas Chair of Advisory Committee	Dr. Rudra Dutta

To my parents . . .

Biography

Jing Teng was born in Wuhan, Hubei province, P.R.China in 1975. He received his B.S. in Computer Science from Wuhan University in 1997, and the M.S. in Computer Science from the Chinese Academy of Sciences in 2000. In 2000, he joined the Department of Computer Science at NC State University to pursue a Ph.D. degree.

Acknowledgements

This work is supported by the Intelligence Technology Innovation Center under contract MDA904-00-C-2133, and by the `Jumpstart` project of MCNC. Also, I am very grateful to members of my advisory committee, Dr. Rudra Dutta, Dr. Khaled Harfoush, Dr. George N. Rouskas, and Dr. Carla D. Savage, especially to my advisor, Dr. George N. Rouksas, for his support and invaluable guidance and suggestion.

Contents

List of Figures

List of Tables

Chapter 1

Introduction on Optical Burst Switching Technology

`Optical Networks`, with their high transmission speed, are the ideal communication infrastructure to meet the continuously increasing demand on bandwidth. In optical networks, the most promising technology so far is *wavelength division multiplexing* (WDM). WDM is essentially the same as the old and widely-used frequency division multiplexing. It is a technology to increase the transmission capacity by transmitting data simultaneously at multiple carrier wavelengths (equivalently, frequencies) [35]. Current technology allows the multiplexing of more than 160 wavelengths at 10 Gbps each to get a total throughput of 1.6 Terabits per second per fiber.

1.1 Optical Burst Switched Networks

Several different technologies have been developed for the transfer of data over WDM networks. Broadcast and select networks, and circuit switched wavelength routing networks have already been extensively studied, and deployed [35]. In broadcast and select networks, signals sent from each node are received by all the other nodes, like most local area networks (LAN), such as Ethernet and Token Ring. This architecture is rather simple, since no routing function is needed. However, it is quite inflexbile, as well as inefficient. Because the signal is broadcast all over the network, it will propagate to parts of the network where it does not need to go, and cause a waste of network resources. Circuit-switched wavelength routing networks are more efficient, where optical signals are selectively transmitted only over part of the network. In this network architecture, an end-to-end all-optical circuit (lightpath) is established before the transmission of data signals, similar to a telephone line in public switched telephone network (PSTN). This network represents the most practical technology in optical networks today.

However, all-optical circuits tend to be inefficient for traffic that has not been groomed or statistically multiplexed, and are wasteful when the sustained traffic volume does not consume a full wavelength. Optical packet switching (OPS) is a technology to transmit user data by the means of optical packets, in which a wavelength is only allocated to a packet when it is transmitted, and it can be reused by others after the transmission. OPS is more efficient and it can be used to support IP traffic and ATM traffic. Readers are referred to [48] for details. However, OPS requires practical, cost-effective, and scalable implementations of optical buffering and optical header processing, which are several years away.

Optical burst switching (OBS) is a technology positioned between wavelength routing (i.e., circuit switching) and optical packet switching. OBS is a technical compromise that may not require optical buffering or packet-level parsing, so it is more practical than OPS, and it is more efficient than circuit switching since the wavelength is only allocated to a burst when it is in transmission. Optical burst switching is an adaptation of ATM block transfer (ABT), which is the standard for burst switching in ATM by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T).

In OBS, the transmission of each burst is preceded by the transmission of a control packet, whose purpose is to inform each intermediate node of the upcoming data burst so that it can configure its switching fabric in order to switch the burst to the appropriate output port. In OBS, as in ABT, two schemes can be used, tell and go (TAG) and tell and wait (TAW). In TAG, immediately after the source transmits the control packet, it transmits the optical burst. This scheme requires optical buffers in the optical switches to store the burst while it waits for the processing of the control packet and the setup of the switching fabric. In TAW, the source does not transmit the burst until it receives the confirmation from the destination, and this two-way reservation is similar to establishing a lightpath in wavelength routing WDM networks. However, in OBS there is a more efficient way to reserve wavelengths. The source node does not wait for the confirmation that an end-to-end connection has been set-up; instead it starts transmitting a data burst after an appropriate delay (referred to as **offset**), following the transmission of the control packet. The delay (offset) should gurantee the control packet has been processed by all the intermediate switches and the switching fabrics of all the switches have been setup when the burst arrives in those switches.

1.2 Related work on Optical Burst Switching

Over the last five years, research in OBS networks has rapidly progressed from purely theoretical investigations [33, 39] to prototypes and proof-of-concept demonstrations [5, 4]. The breadth and depth of current OBS research is evident in the wide range of topics covered during the highly successful OBS workshops which took place in late 2003 alongside the Opticomm and Globecom conferences; the interested reader is referred to the workshop proceedings, available on-line [3].

Studies of OBS networks can be broadly classified as follows.

- **Network architecture.** Reference [39] proposed the idea of optical burst switching, as well as the concept of control channel, data burst, and head cell. It also described a burst switch architecture design as well as its operation. References [10, 46] proposed a network architecture such that OBS network is the backbone of Internet (IP network), and described the architecture of an optical switch (router) with the functions including burst control packet detecting, processing, and rewriting; forwarding table lookup; management of shared buffer (fiber delay line, or FDL); wavelength conversion; and data channel scheduling. Reference [51] studied access schemes in OBS for a metro ring architecture. As the idea of OBS is becoming more mature, reference [9] discussed technical issues and general requirements for a transport layer architecture (i.e., services and protocols) for OBS networks.

- **Resource reservation schemes and protocols.** The manner in which resources are reserved has a significant effect on the performance of an OBS network, and is one of the most basic problems in OBS. A number of wavelength reservation schemes have been proposed for OBS, including just-enough-time (JET) [33], Horizon [39], and just-in-time (JIT) [43]. They have attracted considerable attention in both academia and industry, and will be addressed in detail in Chapter 2. Other reservation schemes include wavelength-routed OBS [20] which uses two-way reservations, and forward resource reservation [31] whose objective is to reduce the burst delay at the edge switches and increase the successful forward reservation probability. Based on the JIT reservation scheme, the `Jumpstart` JIT signaling protocol [5, 6] is an open and published signaling specification in OBS. To improve the performance of the network, references [47, 54] proposed more advanced scheduling algorithms to use the wavelength resources more efficiently. In [56, 18], a new reservation scheme was studied based on JET, where different offset time is assigned to bursts with different quality of service (QoS) requirements where QoS is defined in terms of burst blocking probability.

- **Contention resolution.** In OBS, when one-way reservation schemes (e.g. JIT, JET, and Horizon) are adopted, it is possible that two bursts compete for the same output port, thus contention may arise. To resolve burst contention, a switch has several options: using FDL to buffer one of the bursts; deflecting one burst through another output port [25, 59]; using the wavelength dimension (with wavelength conversion capability); or discarding one burst. When two bursts collide, and a decision is made to discard one burst, instead of dropping the whole burst, the switch may just drop part of the burst, either the last part of the burst currently in transmission, or the first part of the coming burst. This is called burst segmentation [41, 38], or optical composite burst switching (OCBS) [17]. Another approach is to minimize the occurence of contention by using traffic engineering mechanisms at the source [30], an approach that will be addressed in Chapter 4.

- **Burst assembly.** An optical burst usually contains a number of user packets (such as IP packets). Thus the algorithms for burst assembly are very important, as they determine the traffic pattern at the burst level. Reference [22] proposed a simple assembly algorithm which reduces the self-similarity of traffic in an IP backbone network. Reference [11] proposed three assembly algorithms and the results showed that burst assembly can reduce the self-similarity of traffic, and therefore reduce the contention in the core network and make TCP traffic smoother at short time scales.

- **Network model and performance evaluation.** As a burst may contain several user packets, the loss of a burst is critical in OBS networks, which may incur severe problems at upper layers, e.g., TCP packets out of sequence. The performance of OBS networks in terms of burst loss probability has been studied extensively [39, 44, 12, 19, 18, 40, 56] using either simulation or simple analytical models. An output port of an OBS node is analyzed assuming Poisson arrivals and no buffering in [12, 19, 18, 40]. Under these assumptions, an output port can be modeled by a finite number of servers with no queue, each representing a wavelength. In this way, the probability that a burst destined to this output port is lost can be obtained from the Erlang B formula. Using a Poisson model, reference [36] analyzes the burst blocking probability of the whole network and reference [59] analyzes the burst blocking probability when deflection routing is adopted. An output port can also be modeled by an $M/M/m/K$ queue with the assumption of Poisson arrivals and buffering [39, 56], where m is the number of wavelengths and $K - m$ is the capacity of the buffer. A model for multiple classes of bursts is developed in [44]. References [22, 53] studied the assembled traffic for burst arrival patterns and burst length distributions other than Poisson and exponential distribution, respectively, while traffic shaping at the edge of the network was studied in [40, 12]. To reflect the self-similarity of burst traffic, Pareto distribution is used extensively in simulations, for example in [30]. Another model, a 3-state Markovian burst arrival process was studied in [49, 50] and algorithms were designed for theoretical analysis.

- **Protection and restoration.** There are many studies on the protection of circuit switched wavelength routing networks. However, there have been very few works on protection for OBS networks. Reference [29] analyzed survivability of OBS against a single link failure by analyzing the QoS in terms of both burst blocking probability and maximum increment of burst scheduling delay under the failure. Reference [28] proposed a failure-compensating scheduling in OBS nodes, while reference [24] extended $1 + 1$ protection in wavelength routing networks to OBS.

1.3 The `Jumpstart` Project

Since late 2000, NCSU has been collaborating with MCNC-RDI to build a proof-of-concept OBS implementation under the ARDA-funded `Jumpstart` project [1]. (ARDA focuses on high-performance data communications requirements that cannot be addressed by technologies used in today's Internet [2].) NCSU and MCNC-RDI have developed an open, published specification of the `Jumpstart` JIT signaling protocol [5, 6], inspired by an earlier work by Wei and McFarland [43]. 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 (JITPAC) network controllers which implement the signaling protocol in FPGA, and deployed them at three ATDNet sites in November 2002 for experimentation and testing [4]. 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 impacts 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. This is a topic we address in Chapter 2.

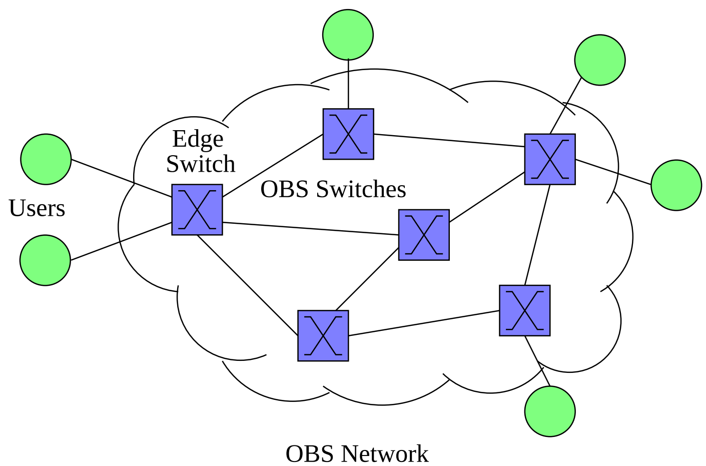

Figure 1.1: An OBS network

1.4 The Optical Burst Switched Network under study

We consider an OBS network consisting of burst switches (also called nodes in this thesis) interconnected by bidirectional fiber links, as shown in Figure 1.1. 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 to transmit control packets, and the other W wavelengths are used to transmit data bursts. We assume that a user is capable of transmitting and receiving on any of these $W+1$ wavelengths simultaneously. In other words, we assume that users are equipped with $W+1$ pairs of optical transceivers, each pair fixed-tuned to one of the wavelengths. We should note here that the terms “user” in OBS does not mean a terminal, or a PC, which can send packets to a router. Instead, it refers to an assembly device that can receive packets from terminals and then assemble them into bursts. This is because a burst is often much larger than an IP packet, or a packet, frame or datagram of other networks that OBS is supposed to support.

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. Each OBS node in the network consists of two main components, as illustrated in Figure 1.1:

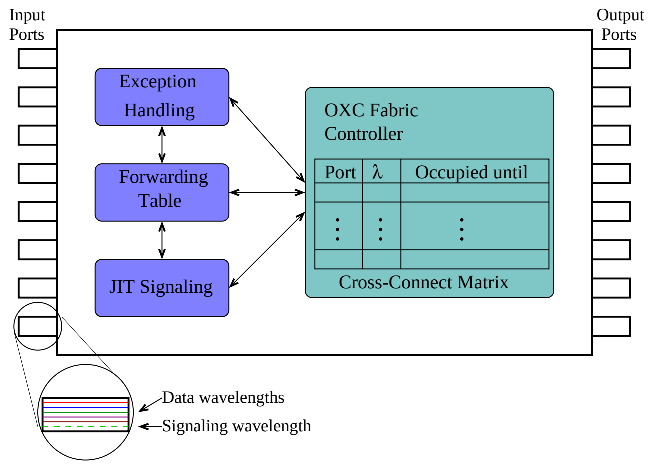

Figure 1.2: Logical diagram of the signaling engine at an OBS node

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, which was developed at MCNC as part of the `JumpStart` protocol, and which has been deployed in the ATDNet, implements the signaling engine in FPGA.) The logical diagram of the signaling engine is shown in Figure 1.2.

2. An *optical cross-connect* (OXC), which performs the switching of bursts from input to output. Figure 1.3 shows the OXC architecture we consider in this study.

In this thesis, we assume that OBS nodes do 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 OBS nodes, the signaling wavelength is always terminated at each 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 [57] of the JIT protocol). Processing a signaling message may involve one or more actions, including:

- The determination of a next hop switch for a burst. We assume that the signaling engine performs a forwarding table lookup to determine the next hop switch for a burst

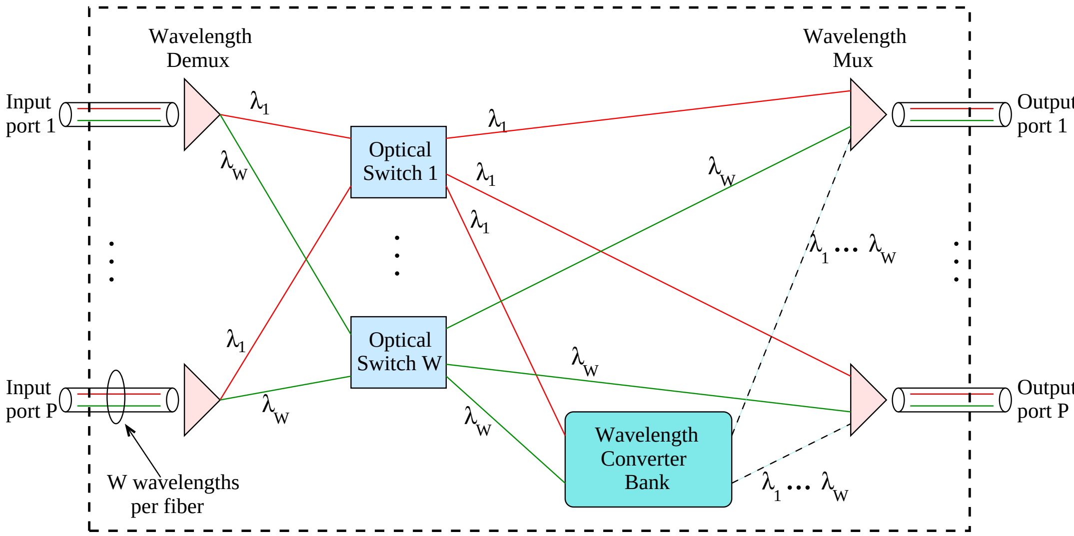

Figure 1.3: A $P \times P$ OXC with a shared converter bank

(and associated `setup` message). The forwarding table is maintained and updated by a routing protocol which is outside the scope of this thesis.

- The forwarding of signaling messages to upstream or downstream nodes.
- The configuration of the OXC switching elements to optically switch bursts from an input to an output port. OXC configuration is accomplished through the OXC fabric controller which keeps track of the state of output wavelength-port pairs. If the immediate reservation is adopted, cross-connect elements are not reserved for future bursts, no scheduling takes place within the OBS node, and the cross-connect matrix is very simple, as shown in Figure 1.2. Otherwise, complicated channel scheduling should be performed.
- The handling of exception conditions.

Figure 1.3 shows the logical diagram of an OXC with P input and P output ports, and W burst wavelengths per fiber. At each input port, the incoming light signal is demultiplexed into its individual wavelengths, which are then passed on to optical switches. The OXC has W optical switches, one for each burst wavelength. We assume that each optical switch consists of a non-blocking space-division switch fabric, with no optical buffers.

The switch fabric may use MEMS micro-mirror arrays as the cross-connect elements [14], or it may employ some other suitable switching technology.

The OXC may provide a wavelength conversion capability. In this case, we assume that the OXC is equipped with a *wavelength converter bank*, consisting of C converters. Each converter is capable of shifting any of the W incoming wavelengths to any of the W outgoing wavelengths. The converter bank is shared among all output ports, as shown in Figure 1.3, since this architecture has been shown to provide good performance in a cost-effective manner.

In order to describe the switching operation of the OXC, consider a `setup` message arriving at some OBS node and announcing the imminent arrival of a burst on some wavelength w on input port x. The signaling engine looks up its forwarding table and determines the output port, say y, for the burst. It then consults the cross-connect matrix to determine the status of wavelength w at output port y. If the wavelength is free, it instructs the OXC fabric controller to perform a connection from input port x to output port y on wavelength w, and it updates the cross-connect matrix. As a result, the optical switch corresponding to wavelength w directly switches the incoming burst (e.g., by raising the appropriate micro-mirror) from input to output.

Consider now the case where wavelength w on output port y is busy transmitting some other burst. If the OXC offers no conversion capability, then the new burst coming in port x is dropped. Suppose now that the OXC is equipped with a bank of C converters shared among the output ports, as Figure 1.3 illustrates. In this case, the status of the converters is checked. If all converters are in use by other bursts, or if all wavelengths of output port y are busy, then the new burst is dropped. Otherwise, let w' be a wavelength that is free on output port y. Then, the OXC fabric controller instructs the optical switch corresponding to wavelength w to switch input x to the wavelength converter bank. One of the free converters in the bank is tapped to converter the incoming signal to wavelength w', and the signal is then switched to the desired output port y.

We now define three important system parameters that we will use in our study.

- T_{OXC} denotes the amount of time it takes the OXC to configure its switch fabric to setup a connection from an input port to an output port. In other words, T_{OXC} is the delay incurred between the instance the OXC receives a command from the signaling engine to set up a connection from an input port to an output port, until the instance 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 the configuration delay is largely independent of the pair of input/output ports that must be connected, as well as of the status 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. Therefore, we take T_{OXC} as a constant in our study.

- $T_{setup}(X)$ 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 will explain in Chapter 2, 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 $T_{setup}(X)$ 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 [58], and thus, the processing time can be bounded.

- $T_{offset}(X)$ is the offset value of a burst under reservation scheme X. It is the amount of time that the signaling message (for reservation) of a burst arrives at a switch ahead of the data. The offset value depends on (1) the OBS protocol, (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 or congestion control [56]. 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:

$$T_{offset}^{(min)}(X) \;=\; kT_{setup}(X) \;+\; T_{OXC} \tag{1.1}$$

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.

This thesis is organized as follows. In Chapter 2 we provide a detailed description of the JIT, JET, and Horizon wavelength reservation schemes, discuss issues related to their hardware implementation, and introduce a new reservation scheme called JIT^{+}. Also we develop analytical models of a single OBS node that capture the performance of the four reservation schemes. Then we present numerical results to compare the relative performance of the four schemes, both on a single OBS node and on a path of OBS nodes with cross-traffic, under a wide range of system parameter values that correspond to current and projected technology. In Chapter 3 we study the performance of OBS networks with linear and mesh topologies under the `Jumpstart` JIT signaling protocol. In the study, we consider the effect of several important system parameters including the number of wavelength converters in a switch, and the signal propagation delay between switches. In Chapter 4, we develop a suite of adaptive and non-adaptive wavelength assignment policies for OBS networks. We also apply traffic engineering techniques to reduce wavelength contention through traffic isolation. Finally, we conclude the thesis in Chapter 5.

Chapter 2

Performance Comparison of Wavelength Reservation Schemes

In this chapter, 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 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 chapter is organized as follows. Section 2.1 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 2.2 we develop analytical models of a single OBS node that capture the performance of the four reservation schemes. In Section 2.3 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. In Section 2.4 we propose an analytical model for OBS networks based on previous study. We conclude the chapter in Section 2.5.

2.1 Wavelength Reservation Schemes for OBS Nodes

The various OBS signaling protocols that have appeared in the literature can be broadly classified along three dimensions: (1) the reservation scheme used to hold wavelength resources for bursts, (2) the method in which the offset value is calculated (and the purpose it serves), and (3) the manner in which control is excercised in the network (i.e., distributed or centralized).

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.

2.1.1 Immediate Reservation (JIT)

Immediate reservation, exemplified by the Just-In-Time (JIT) family of OBS protocols [43, 5], 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 Figure 2.1. 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 shows, once the processing of the `setup` message is complete at time $t + T_{setup}$, a wavelength is immediately

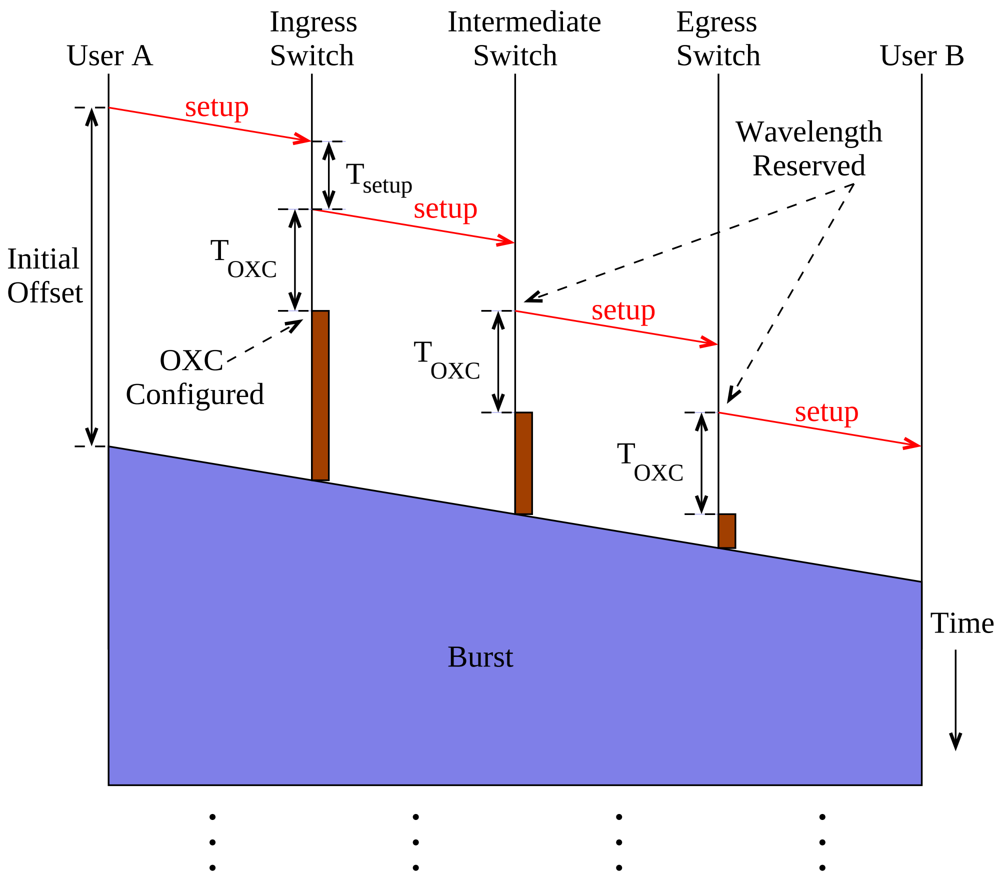

Figure 2.1: Immediate wavelength reservation

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 $t + T_{setup} + T_{OXC}$, 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 $t + T_{offset}$. As a result, the wavelength allocated to the burst remains idle for a period of time equal to $(T_{offset} - T_{setup} - T_{OXC})$. 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.

Figure 2.2 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`. Figure 2.2 shows two successive bursts, i

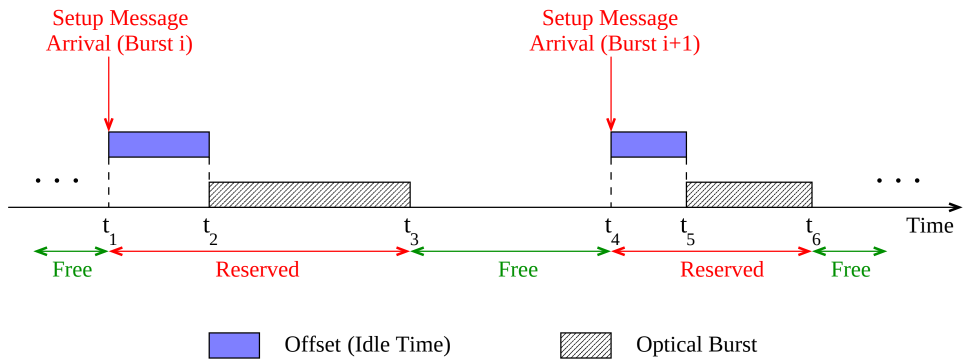

Figure 2.2: Operation and departure process of a wavelength with immediate reservation (JIT)

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 Figure 2.2, the `setup` message corresponding to the i-th 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 transmission ends at time t_6, at which instant the wavelength becomes `free` again.

As Figure 2.2 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.

2.1.2 Delayed Reservation

The Horizon [39] and Just-Enough-Time (JET) [33, 55] 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.*

Figure 2.3 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 $t + T_{offset}$. Assuming that the burst can be accepted, the setup message reserves a wavelength for the burst starting at time $t' = t + T_{offset} - T_{OXC}$. 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.

As we can see in Figure 2.3, a *void* is created on the output wavelength between time $t + T_{setup}$, when the reservation operation for the upcoming burst is completed, and time $t' = t + T_{offset} - T_{OXC}$, when the output wavelength is actually reserved for the burst. If the offset value T_{offset} is equal to the minimum value in expression (1.1), 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.

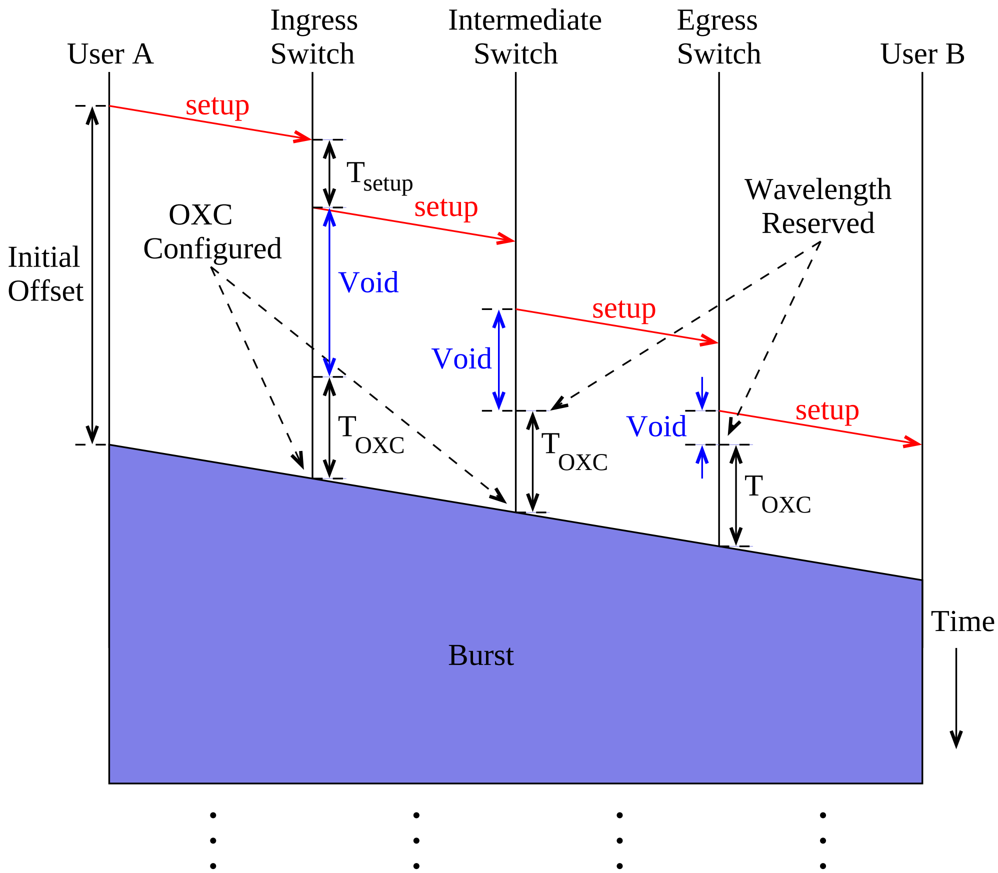

Figure 2.3: Delayed reservation

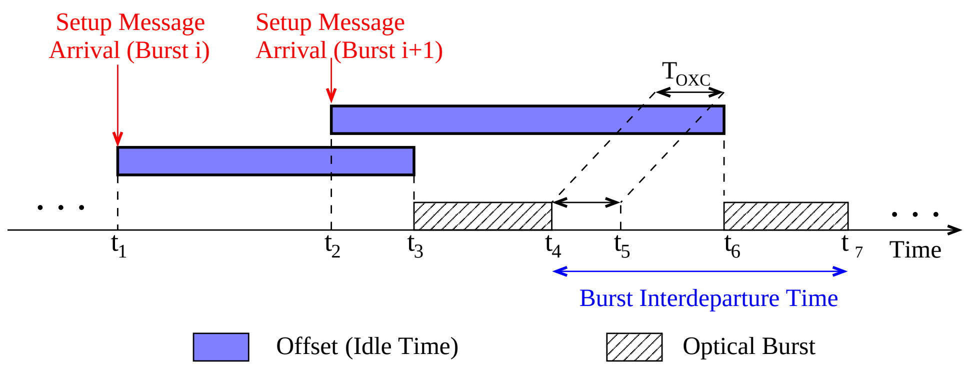

Figure 2.4: Departure process of a wavelength with delayed reservation and no void filling (Horizon)

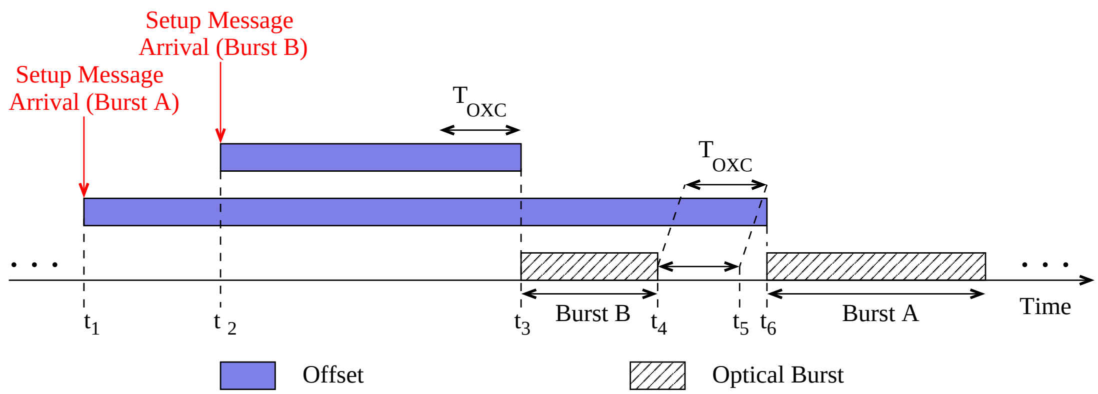

Figure 2.5: Non-FCFS service of a wavelength in an OBS node with delayed reservation and void filling (JET)

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.

Delayed Reservation *Without* Void Filling (Horizon)

Delayed reservation schemes, such as Horizon [39], 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 the* `setup` *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 if and only if the first bit of the burst arrives *after* all currently scheduled bursts on this wavelength have departed.

Figure 2.4 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 Figure 2.4 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.

Delayed Reservation *With* Void Filling (JET)

JET [55] 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 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 2.1.3.

Figure 2.5 illustrates the void-filling operation of JET. The figure shows two bursts, A and B, which 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 switch starts to set up its switching fabric for burst A at time $t_6 - T_{OXC}$. 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 [47, 46] 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 Figure 2.5, burst B is accepted, and it completes service before the start of the configuration of switching fabrics for 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.

2.1.3 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 Figure 2.2 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 [5, 4].

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 [39, 47]. 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 [46] requires a sequential search of *all* wavelength reservation lists for each burst; this takes time $O(m)$ [47], 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 [47] which only takes time $O(\log m)$. However, this algorithm requires the OBS node to maintain complex data

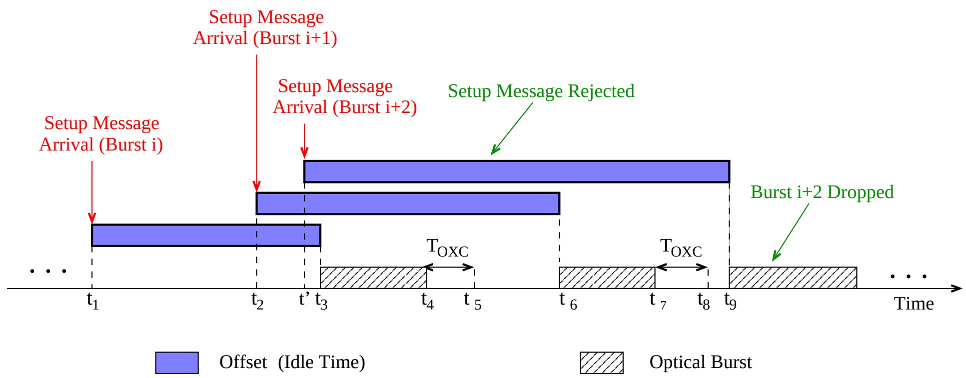

Figure 2.6: Operation of the modified immediate reservation scheme (JIT$^+$)

structures such as red-black trees; therefore, this algorithm is better suited for software, rather than hardware, implementation.

2.1.4 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 2.3), 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.

Figure 2.6 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 Figure 2.4. As in Figure 2.4, 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. 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.

2.2 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 [39, 44, 12, 19, 18, 40, 56]. The

Poisson arrival process is extensively studied in both circuit switched networks and packet switched networks. It is an effective model for telecommunication networks, however, it may not suit for modeling packet switched networks like Internet, especially for wide area traffic [32]. And this problem is even more complicated in OBS since a burst is usually generated by the assembly of a (large) number of packets. So the assembly algorithms have a great influence on the arrival model. In [22, 53, 52], it is pointed out that the input burst traffic to an OBS node might be not Poisson. In some situations, it may approache Gaussian distribution [53], even self-similarity. Thus more sophisticated models need to be discovered and studied. In [50], it is developed a queueing network model that the burst arrival process for an edge OBS node is described by a 3-state Markov process.

However, in this chapter, we assume the burst arrival following a Possion process due to its simplicity. As the number of sources that can generate bursts increase, the burst arrival process will approach a Poisson process[1]. Also, as we describe in Section 2.4, we can assume that the traffic from one node to another is Poisson and this assumption will not have great influence on the performance of the network in terms of burst blocking probability.

- The burst size distribution (service model) is also highly dependent on assembly algorithms. Hence we assume that the burst size has 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 $\overline{T}_{offset}(X)$ 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

[1]We should note here that this Poisson property is caused by the independence of multiple assembly devices, not by the applications that generate packets, which is the case in Internet and which is subject to the properties of traffic generating by different network applications [32].

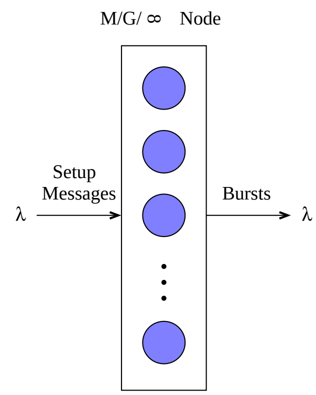

Figure 2.7: Burst arrival process

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 (1.1), 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 Figure 2.7. 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 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 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:

$$Erl(\rho, m) = \frac{\rho^m/m!}{\sum_{i=0}^{m} \rho^i/i!} \tag{2.1}$$

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.

2.2.1 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 Figure 2.2. 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 $\overline{T}_{offset}(JIT)$. 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 $1/\mu + \overline{T}_{offset}(JIT)$.

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 $\rho(JIT)$ of the queue is:

$$\rho(JIT) = \lambda\left(\frac{1}{\mu} + \overline{T}_{offset}(JIT)\right) \tag{2.2}$$

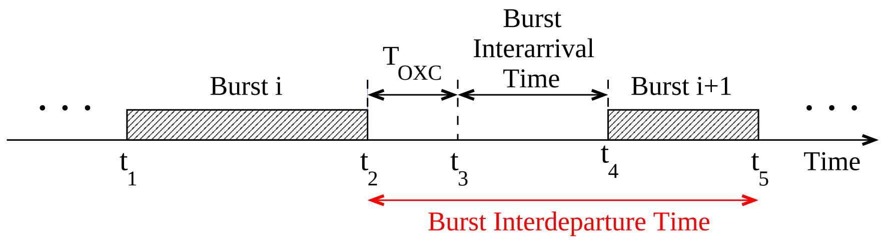

Figure 2.8: Departure process of a wavelength in an OBS node with delayed reservation and void filling (JET)

and the burst drop probability is given by $Erl(\rho(JIT), W)$. 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 [19], where, however, the assumption was made that burst (rather than `setup` message) arrivals are Poisson; in that case, the model is only approximate.

2.2.2 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 [46, 47] when allocating a wavelength to a burst, as in the JET [55] 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 Figure 2.5.

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 Figure 2.8, 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 Figure 2.8 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 $t_3 = t_2 + T_{OXC}$. 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 $B^{\star}(s)e^{-sT_{OXC}}$ and mean $1/\mu + T_{OXC}$.

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 $\rho(JET)$ for this system is given by

$$\rho(JET) \quad = \quad \lambda\left(\frac{1}{\mu} \; + \; \overline{T}_{OXC}\right) \tag{2.3}$$

and the probability of burst loss at the output port is given by the Erlang-B formula $Erl(\rho(JET), W)$. 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 [19] (as well as other studies) does not include the term T_{OXC}, resulting in a lower value than the one in (2.3). Since these studies ignore the OXC configuration time, their results underestimate the burst loss probability of JET.

2.2.3 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, $1/\mu + T_{OXC}$, *plus* a quantity $\Delta \geq 0$. In other words, we use the following value for the traffic intensity of Horizon:

$$\rho(Horizon) \quad = \quad \lambda\left(\frac{1}{\mu} \;+\; \overline{T}_{OXC} \;+\; \Delta\right) \tag{2.4}$$

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 our discussion in Section 2.2.5), 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 (2.3)), resulting in higher burst drop probability. Using $\Delta > 0$ in (2.4) 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 (2.4). Finally, we note that some previous studies, including [19], 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.

2.2.4 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 (2.4), 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$^+$.

2.2.5 Discussion

If we ignore the differences in the `setup` message processing time $T_{setup}(X)$ among the four 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 three 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.

- $T_{OXC} > kT_{setup} \Rightarrow$ JET $\equiv$ Horizon $\equiv$ JIT$^+$

  Referring to (1.1), if T_{OXC} is larger than the sum of `setup` message processing times, then no void filling may take place. This is because one OXC configuration operations is needed for a burst with a later `setup` message to fill a void created by a burst with an earlier `setup` message, while the void is at most equal to $kT_{setup} \leq T_{OXC}$. 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 Figure 2.6. 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 $2T_{OXC}$, 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 $+T_{OXC} > kT_{setup} \Rightarrow$ 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$^+$.

- $T_{offset} =$ constant $\Rightarrow$ JET $\equiv$ Horizon

  If the offset value is constant (rather than equal to the minimum value in (1.1)), 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 (1.1), 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.

- $(1/\mu \gg T_{OXC}$ and $1/\mu \gg T_{setup}) \Rightarrow$ JET $\approx$ Horizon $\approx$ JIT$^+$ $\approx$ JIT
  If the mean burst size $1/\mu$ is large relative to the values of T_{OXC} and T_{setup}, then from (1.1), 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 (2.2), (2.3), and (2.4)) 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 (1.1)), 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.

2.3 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 2.3.1) and a path of OBS networks with cross-traffic (Section 2.3.2). For the single OBS node, we use the Erlang-B formula (2.1) 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 used simulation only 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. 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 2.1. Scenarios 1 and 2 in the table correspond to T_{OXC} and $T_{setup}(JIT)$ values that reflect currently available technology. Specifically, we let $T_{OXC} = 10ms$, a value that represents the configuration time of existing MEMS switches [14], and $T_{setup}(JIT) = T_{setup}(JIT^+) = 12.5\mu s$, a value that corresponds to the processing time of JIT signaling messages in our JITPAC controllers [4]. 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, $T_{setup}(JET) = 50\mu s, T_{setup}(Horizon) = 25\mu s$, 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 $1/\mu = 5T_{OXC} = 50ms$, while in Scenario 2 we let $1/\mu = T_{OXC} = 10ms$. 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 2.1 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 $T_{setup}(JIT) = T_{setup}(JIT^+) = 1\ \mu s$ (an improvement of one order of magnitude). These projections assume that the less mature OXC technology will improve faster than the more

Table 2.1: Values of the system parameters for the various traffic scenarios used in the performance comparison

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

mature hardware processing technology. The values of $T_{setup}(JET)$ and $T_{setup}(Horizon)$ relative to $T_{setup}(JIT)$ are the same as above. Also, the difference between Scenario 3 and Scenario 4 is that the mean burst size takes values equal to $5T_{OXC}$ and T_{OXC}, 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 $500ns$, and `setup` processing times for JIT and JIT$^+$ will decrease to $50ns$. 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 (1.1). 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 [46, 47] 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.

2.3.1 A Single OBS Node

The six Figures 2.9-2.14 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 2.1, 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

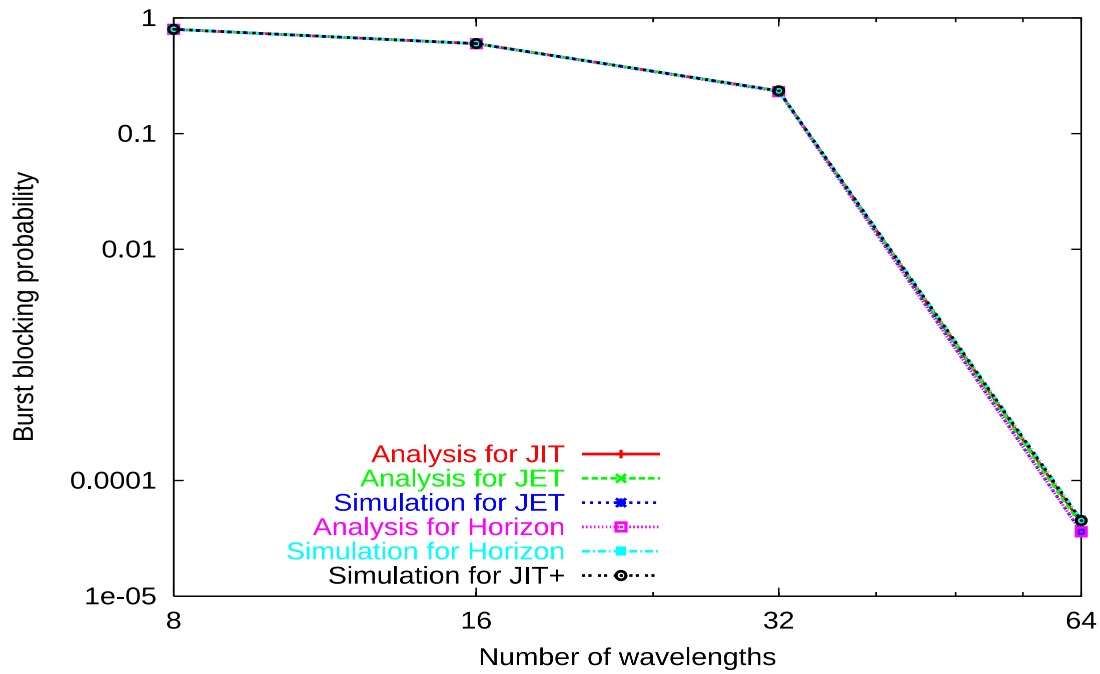

Figure 2.9: Single node performance comparison, Scenario 1 (current technology, $1/\mu = 5T_{OXC}$)

$1/\mu = 5T_{OXC}$, 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 2.9, 2.11, and 2.13), 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 2.10, 2.12, and 2.14). This behavior is due to the high burst switching and `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

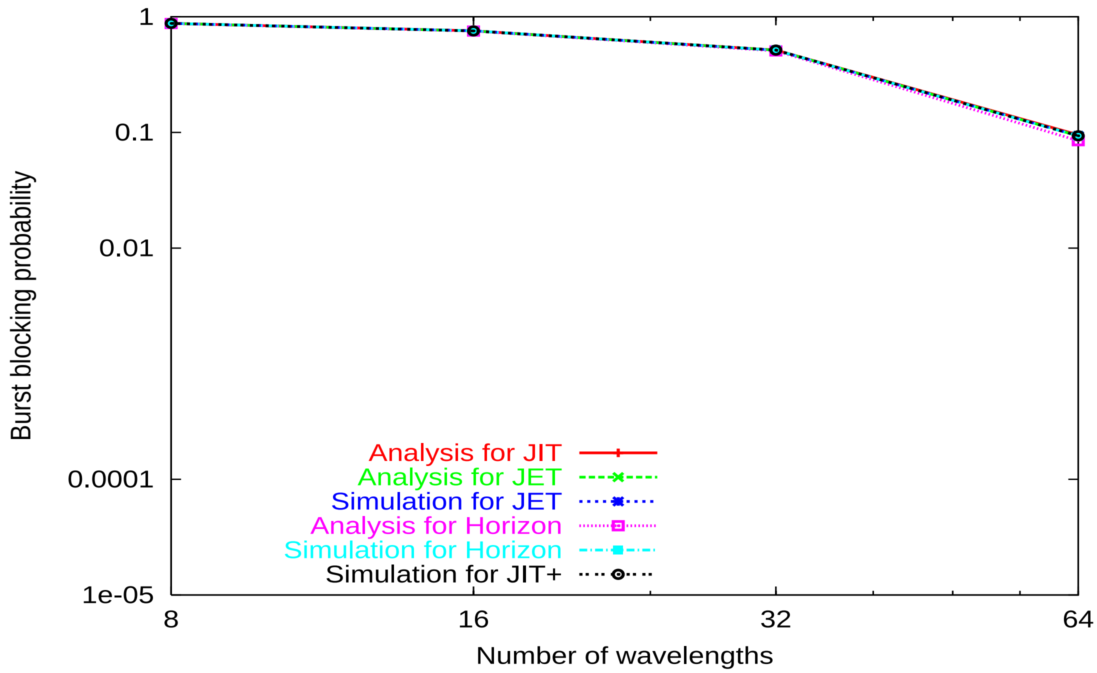

Figure 2.10: Single node performance comparison, Scenario 2 (current technology, $1/\mu = T_{OXC}$)

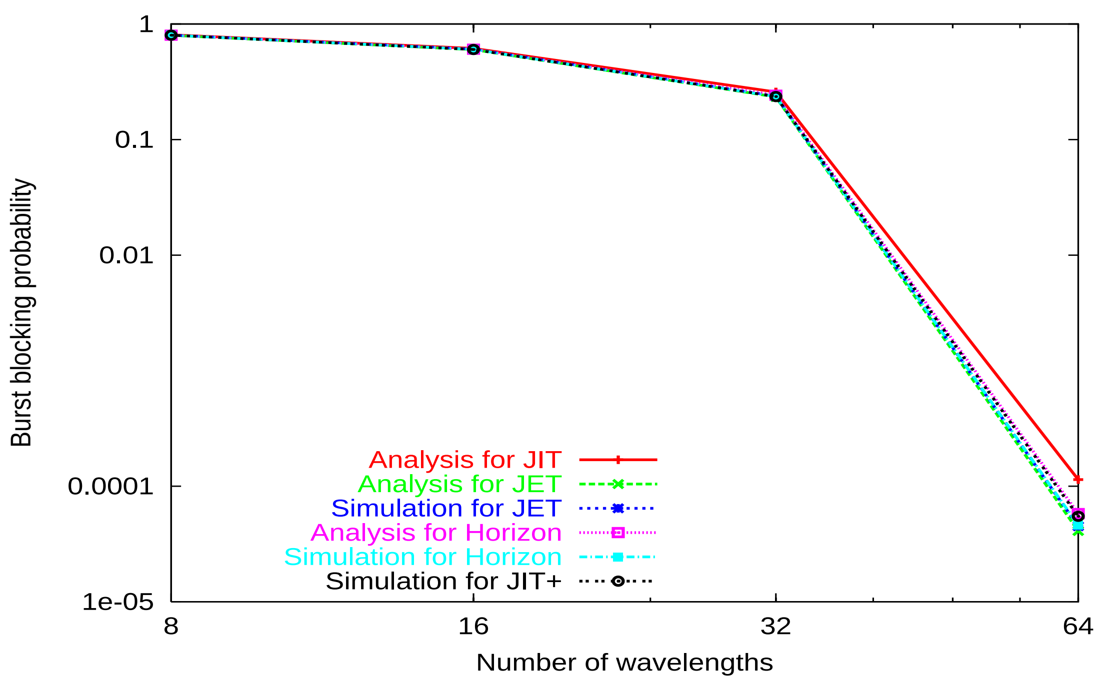

Figure 2.11: Single node performance comparison, Scenario 3 (near future technology, $1/\mu = 5T_{OXC}$)

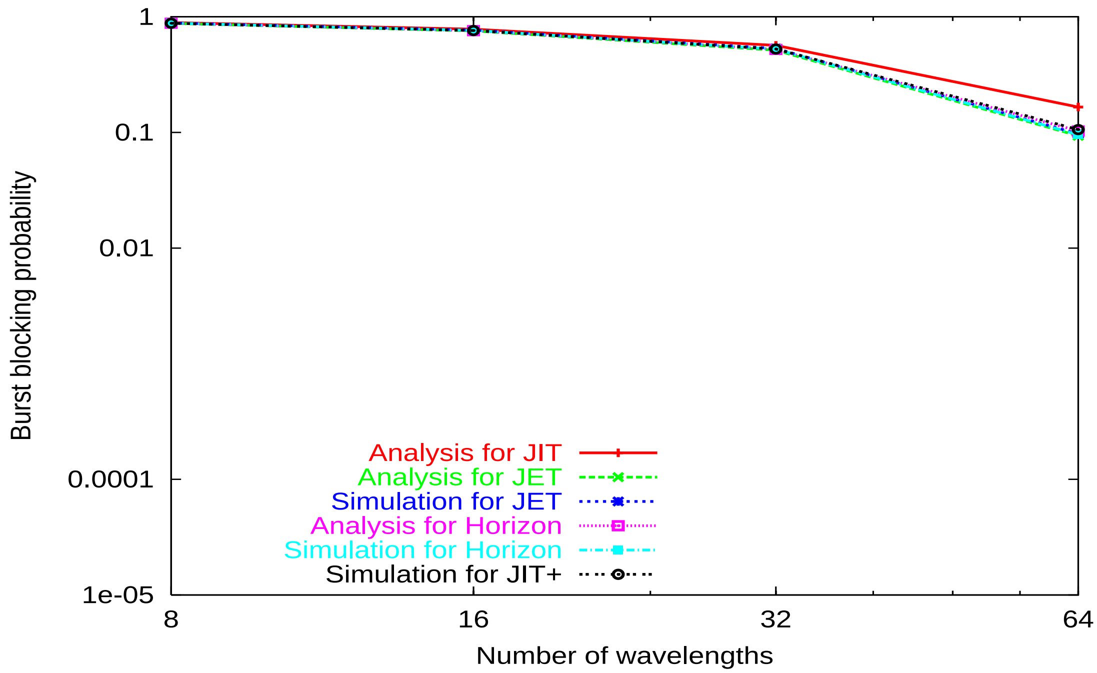

Figure 2.12: Single node performance comparison, Scenario 4 (near future technology, $1/\mu = T_{OXC}$)

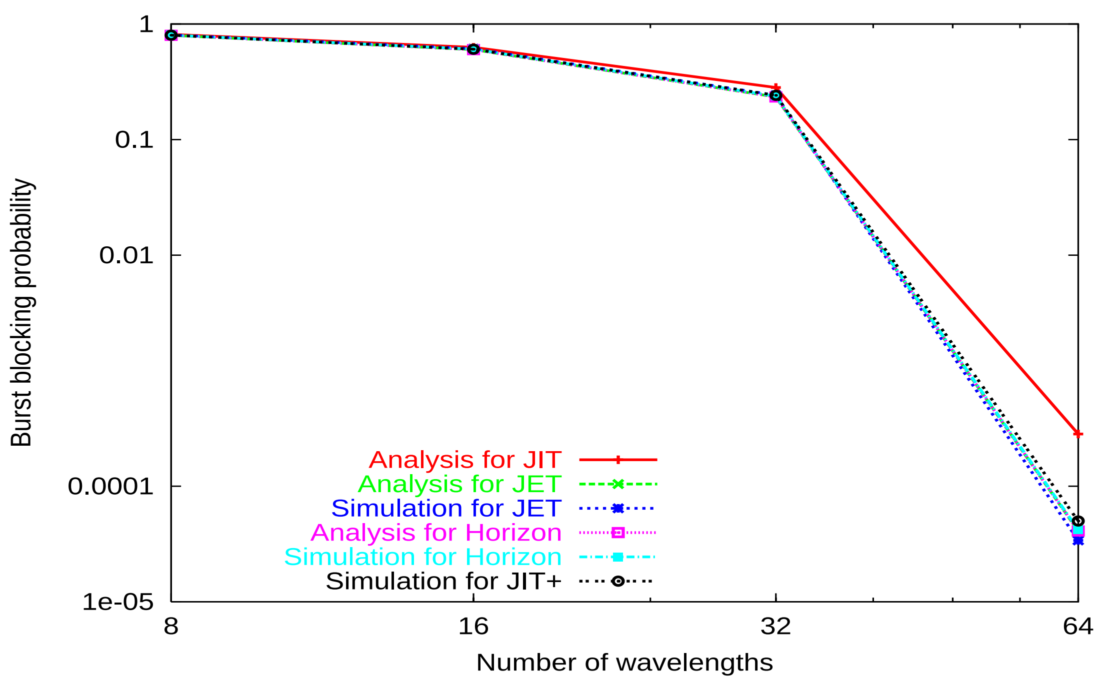

Figure 2.13: Single node performance comparison, Scenario 5 (distant future technology, $1/\mu = 5T_{OXC}$)

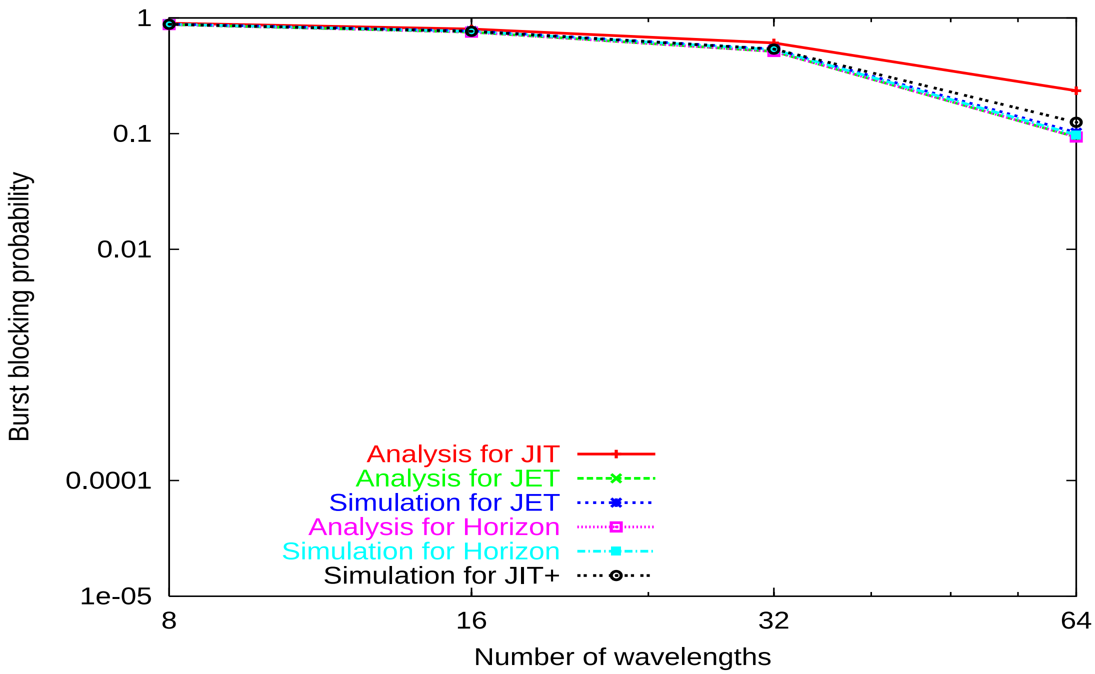

Figure 2.14: Single node performance comparison, Scenario 6 (distant future technology, $1/\mu = T_{OXC}$)

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 system 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 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 2.10, 2.12, and 2.14 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.

2.3.2 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 Figure 2.15. 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_1 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 Figure 2.15. 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 2.1, and we let the arrival rate λ of `setup` messages be such that $\lambda/\mu = 32$ for *all* scenarios.

Figures 2.16-2.21 plot the burst drop probability of the *through traffic* (i.e., traffic from node S_1 to all other nodes), in the path of Figure 2.15; each figure corresponds to one of the scenarios listed in Table 2.1. Note that the burst drop probability in each of

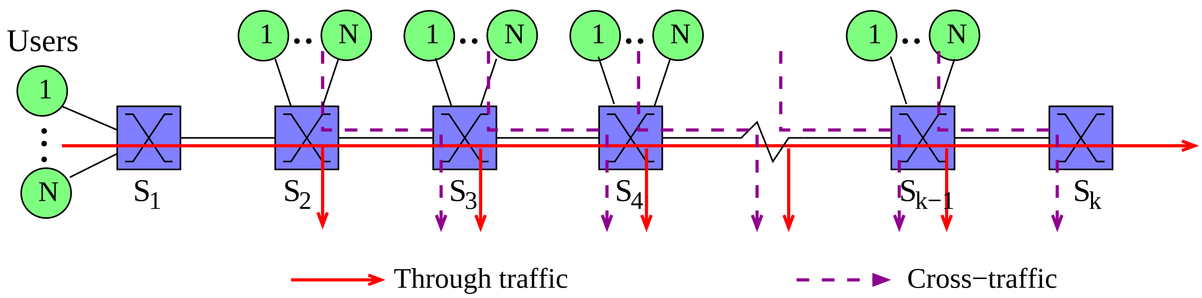

Figure 2.15: The linear OBS network

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 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 [56]. 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 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

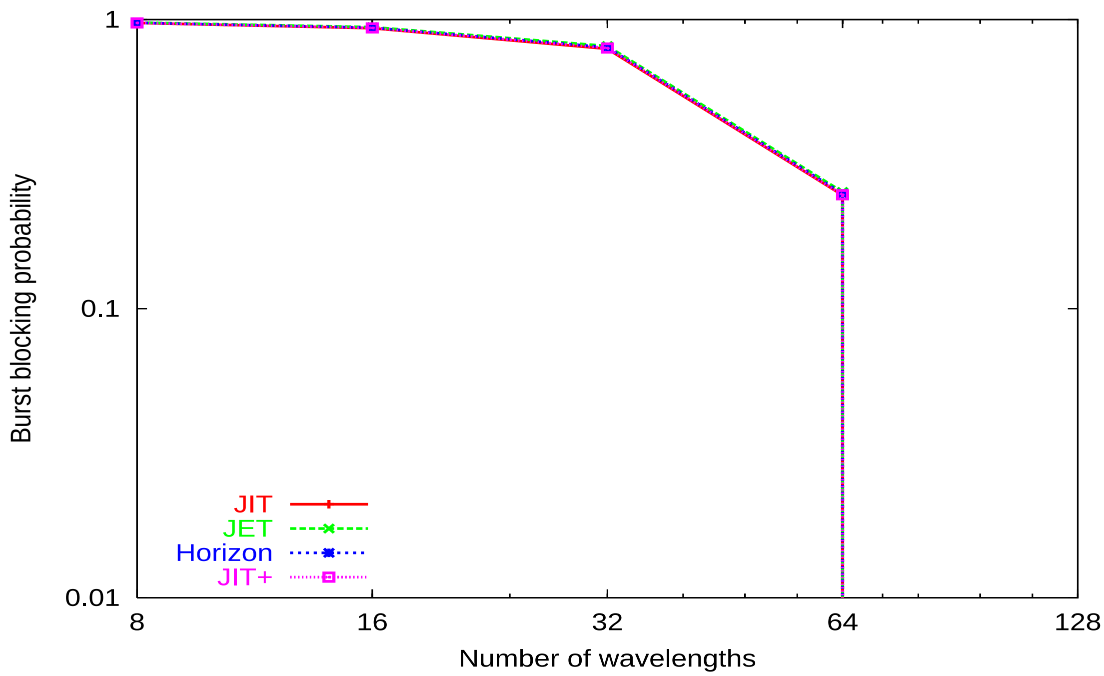

Figure 2.16: Path performance comparison, Scenario 1 (current technology, $1/\mu = 5T_{OXC}$)

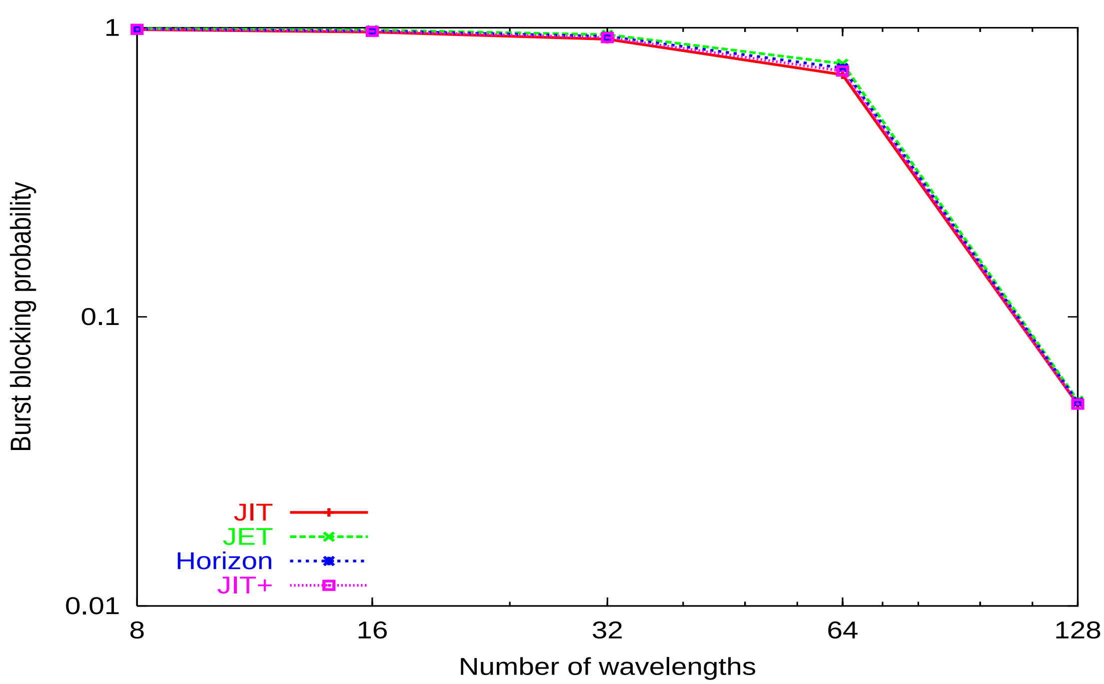

Figure 2.17: Path performance comparison, Scenario 2 (current technology, $1/\mu = T_{OXC}$)

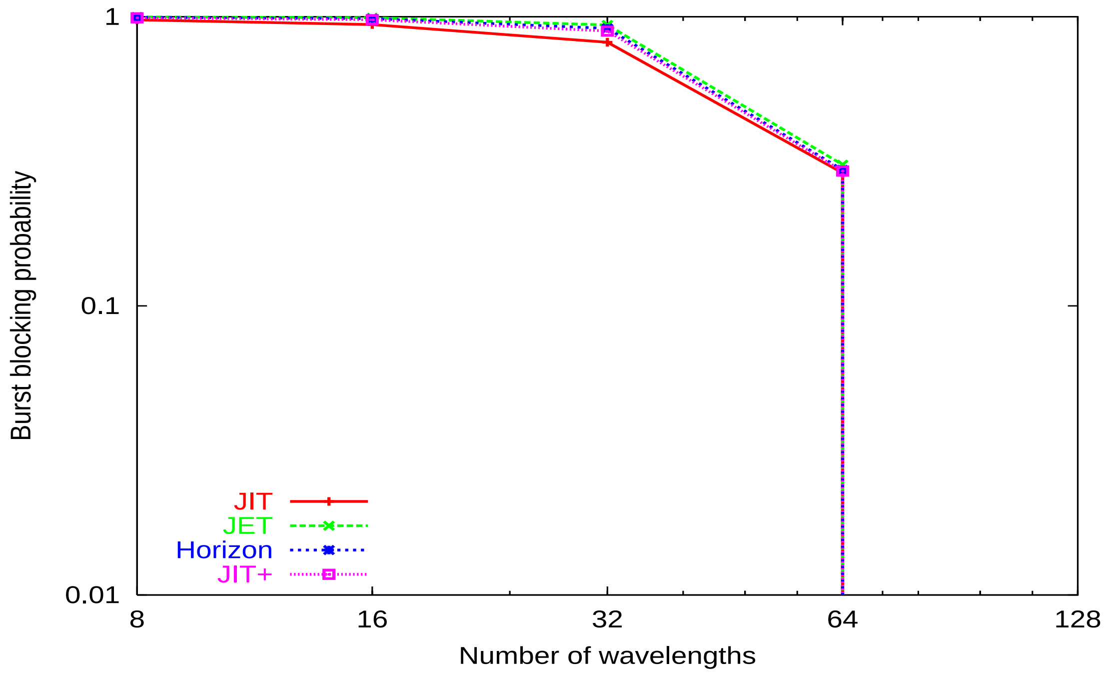

Figure 2.18: Path performance comparison, Scenario 3 (near future technology, $1/\mu = 5T_{OXC}$)

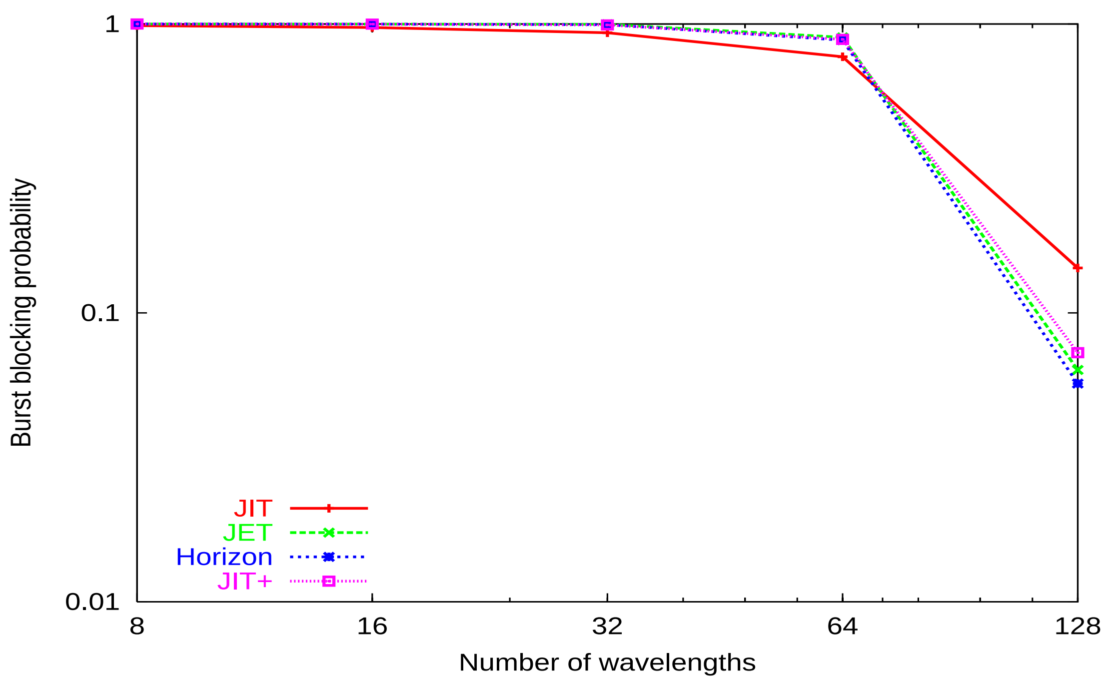

Figure 2.19: Path performance comparison, Scenario 4 (near future technology, $1/\mu = T_{OXC}$)

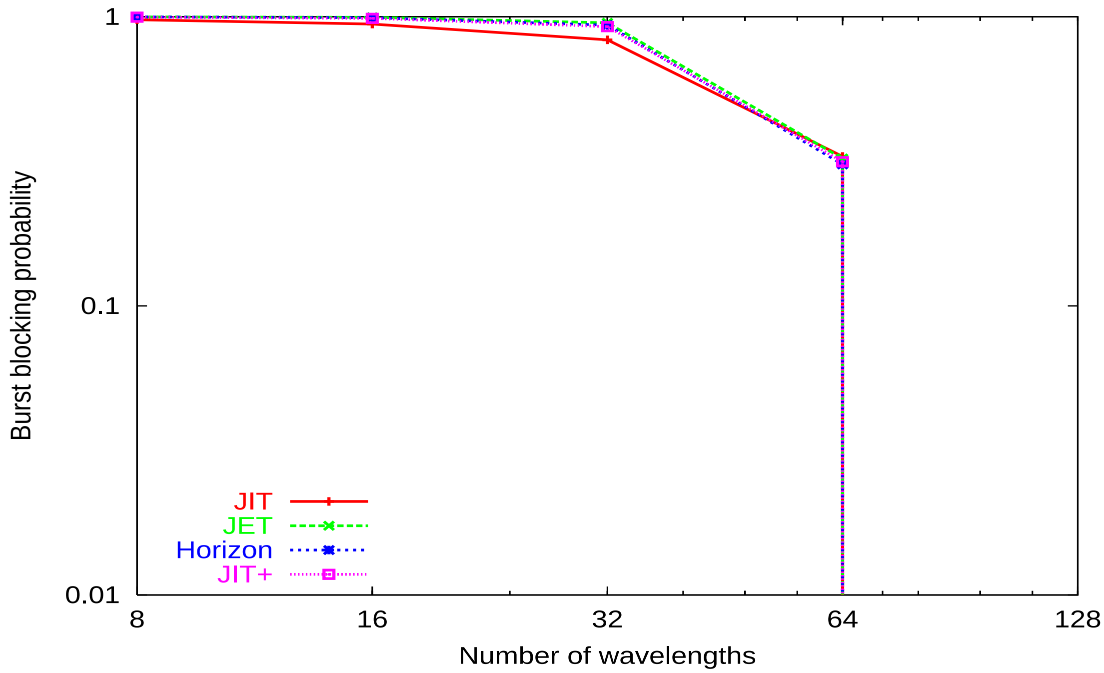

Figure 2.20: Path performance comparison, Scenario 5 (distant future technology, $1/\mu = 5T_{OXC}$)

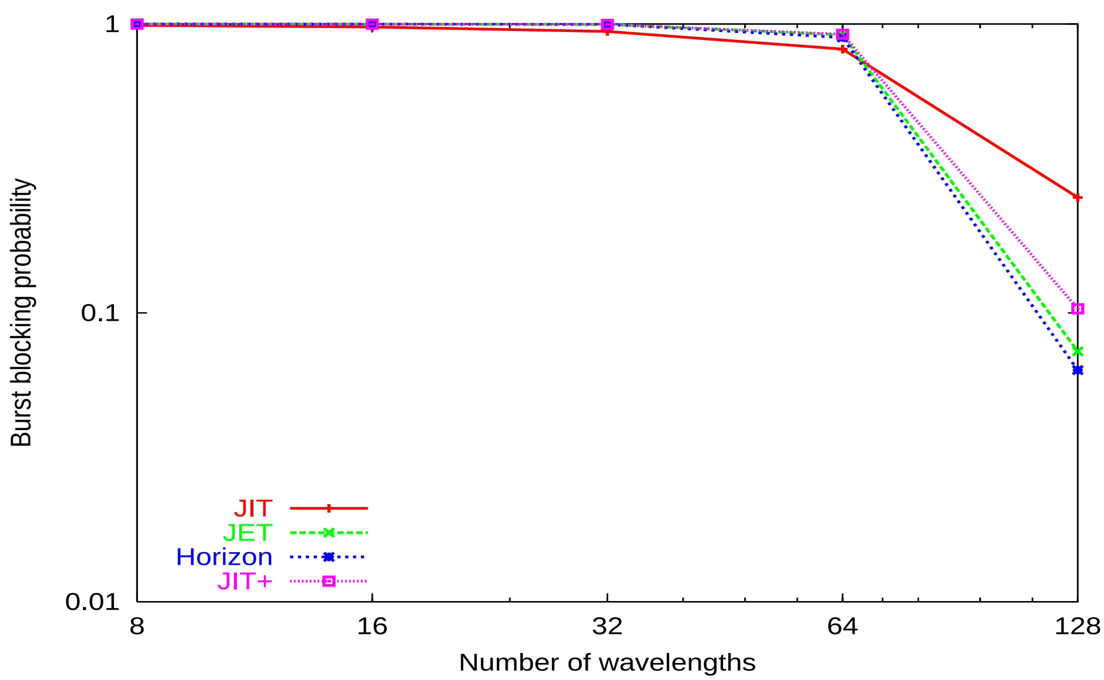

Figure 2.21: Path performance comparison, Scenario 6 (distant future technology, $1/\mu = T_{OXC}$)

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.

2.4 An analytical Model for OBS Networks

OBS networks share properties from both packet-switched networks and circuit-switched networks, making it difficult to analyze using models developed for either of these networks. The difficulty of the analysis is due to the following observations:

1. Though we can assume the overall traffic from users to an edge switch is Poisson, the traffic from one switch to another may not be exactly Poisson.

2. Optical buffering is very expensive and difficult to be implemented.

3. Wavelength continuity constraint. When a burst arrives at a node, it may be dropped if its wavelength in the destined output port is occupied, if no wavelength converter is available at the time.

4. Separation of control and data messages.

5. A burst may occupy multiple links at the same time.

6. In circuit switching, the propagation of signaling messages is typically assumed to be negligible compared to the call connection time. This is not true in OBS networks, thus the propagation delay of signaling messages must be accounted for.

The first five properties distinguish OBS from packet switched networks and the last property distinguishes OBS from circuit switched networks (particularly, wavelength routed networks).

In this section, we develop an analytical model for OBS networks. As we discussed in the previous sections of this chapter, an OBS switch can be modeled by an $M/G/W/W$ node, regardless of what wavelength scheme is used. In the model, we assume the burst arrival time is the arrival time of its `setup` message, and the departure time of the burst is the time that the last bit of data of the burst leaves the node. Further, we assume that there is full wavelength conversion at every node.

Table 2.2: Properties of Output Process of an $M/G/W/W$ node

λ	W	p-value	blocking probability
16	12	< 0.01	0.342421
16	16	< 0.01	0.176174
16	20	< 0.01	0.065708
16	24	> 0.15	0.014603
16	28	> 0.15	0.001857
32	28	< 0.01	0.211796
32	32	< 0.01	0.128790
32	36	0.0256	0.066885
32	40	0.1336	0.024961
32	44	> 0.15	0.007868
32	48	> 0.15	0.001926

Intuitively, an OBS network resembles more a packet-switched network than a circuit-switched network. However, such a network, which consists of $M/G/W/W$ nodes, is very difficult to analyze, because the output process of a node is not Poisson, and we cannot apply those methods developed in a normal packet switched network to analyze it. Fortunately, as the number of wavelengths increases, it is becoming an $M/G/\infty$ node and it is well-known that the departure process of the node is Poisson. So, considering an $M/G/W/W$ node, when the number of wavelengths is sufficiently large, and the burst blocking probability is small, we expect the output traffic to be approximately Poisson[2].

We simulate burst traffic being forwarded by an $M/G/W/W$ node, collecting the results of the burst blocking probability and studying the departure process. As the interval of a Poisson process is exponentially distributed, we use the Kolmogorov-Smirnov method[15] to fit the interval of the departure process with exponential distribution, and p-values are computed to represent the goodness of fit. The results are shown in Table 2.2. The larger the p-value, the better the fit. Particularly, if the p-value is greater than 0.15, we assume it is a good fit, while p-value less than 0.01 represents a bad fit. From Table 2.2, we can find that when burst blocking probabilities are about 0.01 or less, the p-values are greater than 0.15, which means exponential distribution is a good fit for the interval of output bursts. So we can assume the output process is Poisson.

[2]If the burst blocking probability is zero, i.e., the number of wavelengths is always larger than the number of coming bursts and the node is an BCMP type 3 service center [8], the departure process is exactly Poisson.

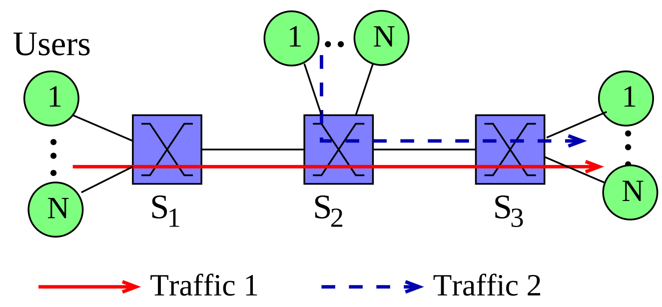

Figure 2.22: A three-node OBS network

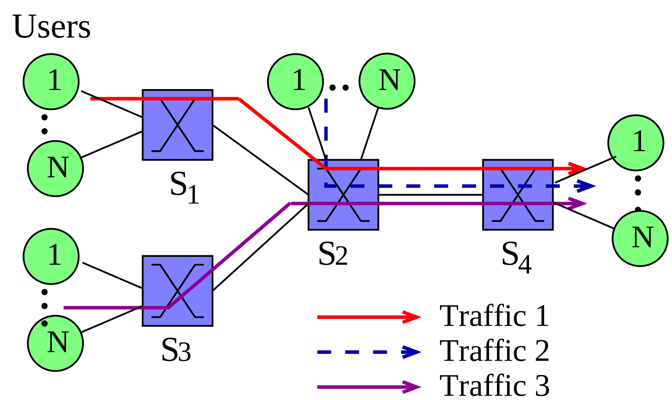

Figure 2.23: A four-node OBS network

Now let us study a three-node OBS network shown in Figure 2.22. There are three nodes S_1, S_2 and S_3, each has a number of users that can generate bursts requests. The traffic from S_1 (passing S_2) to S_3 is called traffic 1, and the traffic from S_2 to S_3 is called traffic 2. The two traffics are both Poisson with arrival rate λ_1 for traffic 1 and $lambda_2$ for traffic 2, respectively.

Let us denote the average effective burst service time is $1/\mu$, $b_{i,j}$ is the burst blocking probability of traffic i in switch S_j. Figure 2.24 shows the burst blocking probability of the network, from both simulation and analysis. In the analysis, $b_{1,1}$ is computed following the Erlang-B formula $Erl(\lambda_1/\mu, W)$, as described in the previous sections. Now we assume traffic 1 to S_2 is also Poisson, and its arrival rate is $\lambda_1' = (1 - b_{1,1})\lambda_1$, then the overall burst blocking probability in S_2 is given by $Erl(\lambda_1'/\mu + \lambda_2/\mu, W)$.

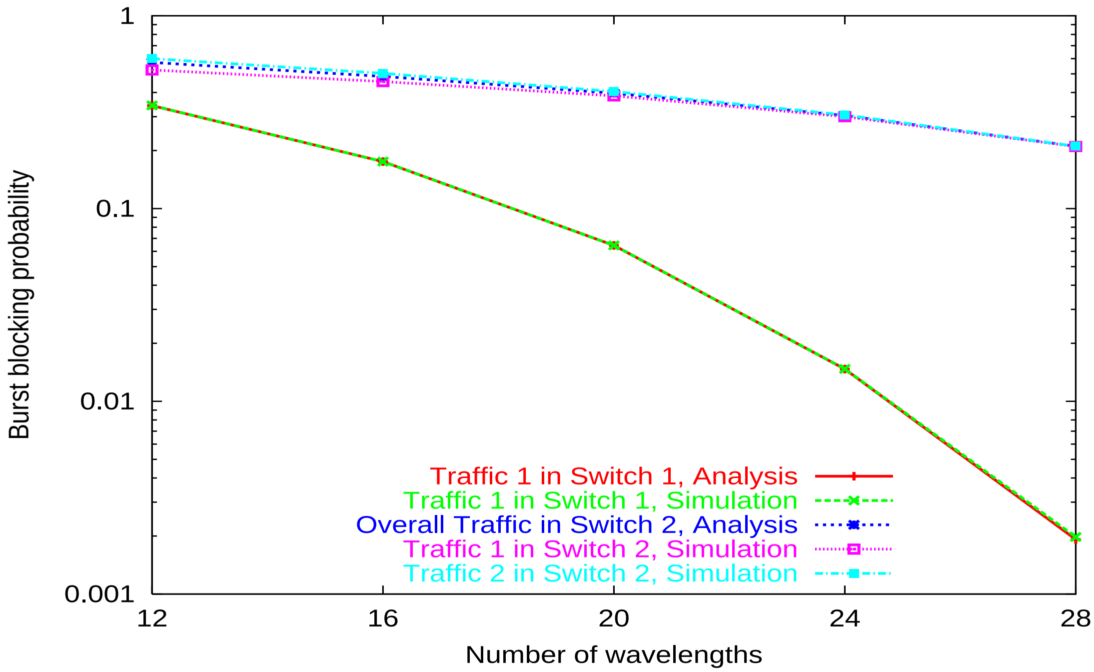

Figure 2.24: Simulation and Analytical result for a three-node OBS network, $\lambda_1 = \lambda_2 = 16\mu$

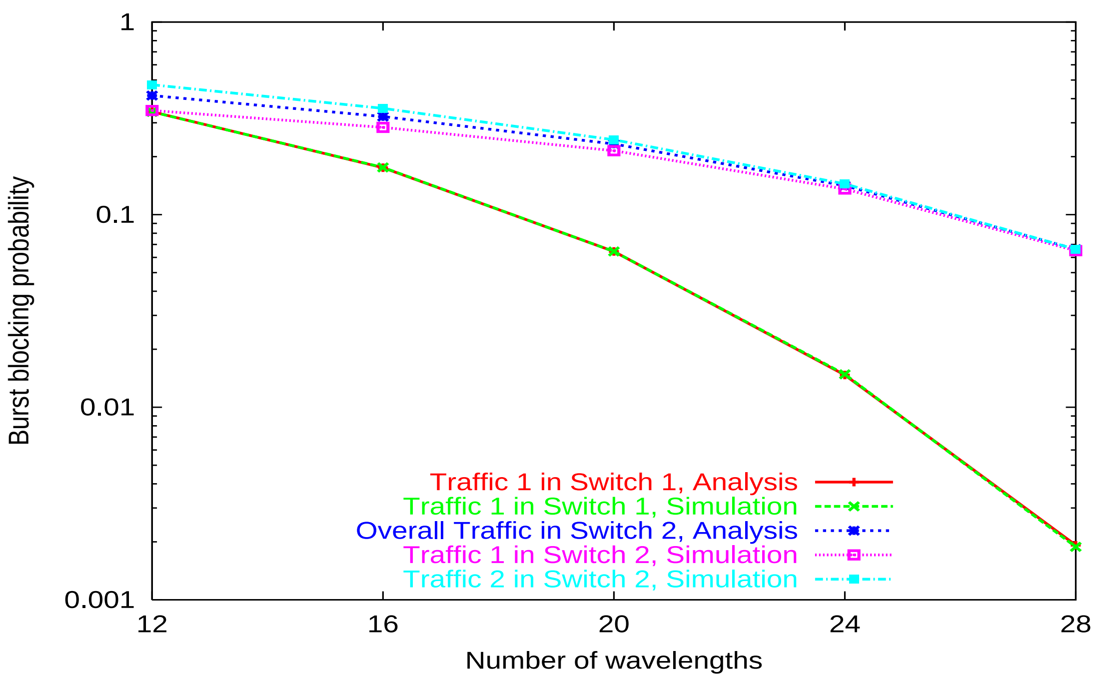

Figure 2.25: Simulation and Analytical result for a three-node OBS network, $\lambda_1 = 16\mu$, $\lambda_2 = 8\mu$

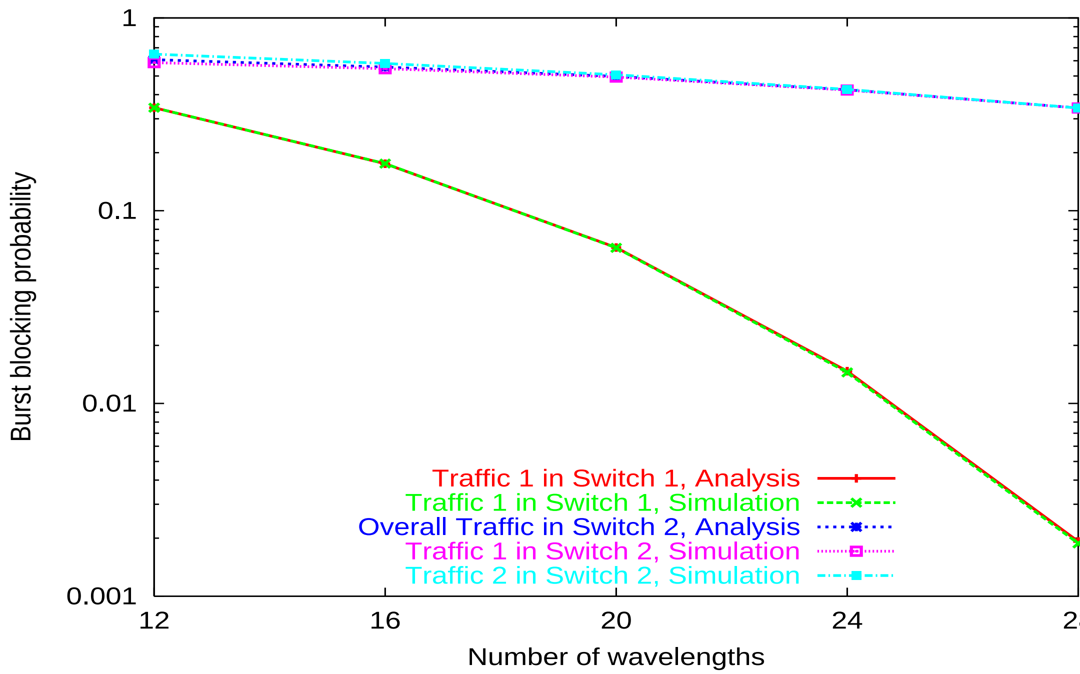

Figure 2.26: Simulation and Analytical result for a four-node OBS network, $\lambda_1 = \lambda_2 = 16\mu$, $\lambda_3 = 8\mu$

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 $160,000$ bursts are transmitted by one switch. 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.

From Figure 2.24, we find the simulation and analytical results of $b_{1,1}$ are very similar, which confirms the results we presented in Section 2.3.1. When we assume traffic 1 to S_2 is Poisson, $b_{1,2}$ and $b_{2,2}$ should be the same, however, when the number of wavelengths is small (less than 24), $b_{2,2}$ is greater than $b_{1,2}$, and the results obtained by analysis is in between.

We credit this phenomenon to the lack of Poisson property of traffic 1 to S_2. (1) Some bursts are dropped at S_1. (2) None pair of bursts of traffic 1 will conflict in wavelength in S_2 (otherwise they should be overlapped transmitted through the same wavelength in S_1, which is impossible), which induces some kind of *dependence* among bursts, and we may expect $b_{1,2}$ is smaller than the analysis result[3], and accordingly, $b_{2,2}$ tends to increase due

[3]Extend to the extremity, suppose $\lambda_2 = 0$, then there will be no burst loss of traffic 1 in S_2

to more wavelength resources occupied by traffic 1.

Figure 2.24 shows the results under the same system parameters, except that $\lambda_2 = 8\mu$. We can see the same trend, except that the gap between $b_{1,2}$ and the results from analysis, as well as the gap between $b_{2,2}$ and the result from analysis, are larger. This is because traffic 1 to S_2 is much heavier than traffic 2 to S_2, so the burst blocking probability of the traffic 1 tends to drop more from the analysis since more traffic can take the advantage from the dependence.

Figure 2.26 shows the results of a four-node OBS network in Figure 2.23. The blocking probability of traffic 3 in S_3 and S_2 are not in the figure since they are the same with those of traffic 1 in S_1 and S_2, respectively. We can find the same trend, and we also find the gaps are smaller than both in Figure 2.24 and Figure 2.25, which means the analytical model fits the simulation better. We think this is because under the presence of traffic 3, traffic 1 to S_2 represents a smaller part of the overall traffic to S_2, which means an increase of independence among the bursts, so bursts of traffic 1 can take less advantage from *no wavelength conflict* with their peers in traffic 1.

Thus we reach the following conclusions:

- If the burst blocking probability is very low in a certain switch, we can assume the output process from this switch is Poisson.

- If none of the traffics from other switches to a certain output port of a switch takes a great part of the overall traffic to the port, we can assume the overall traffic to the port is Poisson.

Fortunately, in practice, it is reasonable to assume there are many users and switches, who will forward bursts to a certain output port of a switch, so it is very unlikely that a certain switch to switch traffic can dominate the overall traffic. And in OBS networks, since a burst will carry a large amount of information, we may want to design the network so that the burst blocking probability is small in every switch. So the two conditions are naturally met. Following our assumption, the burst blocking probability in every output port of every switch can be computed by the link decomposition method in [23].

Finally, we note the limitation of this analysis. First, it assumes the effective service time of a burst will not change too much through the path[4], however, it is not

[4]It does change. When a burst goes deeper into the network, its offset will shrink which might reduce its effective service time in JIT and Horizon. However this change is usually small.

true. For example, if a burst is dropped at one switch, and a `reject` message is forwarded backward to notify the previous switches to release the resources taken by the burst, its effective service time is reduced. Second, in Horizon and JET, bursts of different offsets have different chances to be served [56, 18], so they have different blocking probability. Therefore more research needs to be done to further the analysis of an OBS network.

2.5 Conclusion

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.

Chapter 3

Performance of OBS Networks with the `Jumpstart` JIT Signaling Protocol

In this chapter, we use simulation to study a network of OBS nodes employing the `Jumpstart` signaling protocol. Our objective is to study the performance of networks of various topologies under the influence of important system parameters including the number of wavelength converters, the signal propagation delay between a user and a switch, as well as that between switches. Since, as we discussed in Section 2.4, such networks are generally very difficult to analyze theoretically, we use simulation to study their performance.

3.1 The `JumpStart` Just-In-Time Signaling Protocol

We now discuss the aspects of the `JumpStart` just-in-time (JIT) signaling protocol that are relevant to the simulation model of the OBS network. For a detailed description of the protocol, the interested reader is referred to [5, 6].

Upon the arrival of a burst, a user first sends a `setup` message to its edge OBS

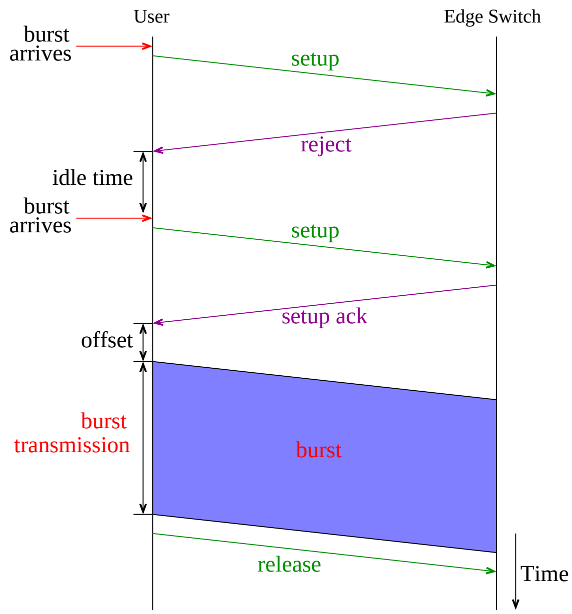

Figure 3.1: Signaling messages between a user and its edge switch in the `JumpStart` JIT protocol

node. The `setup` message carries information related to the burst transmission, including the source and destination addresses. If the edge node can switch the burst, it returns a `setup ack` message to the user. The `setup ack` message contains an offset field that informs the user how long it should wait before transmitting its burst[1]. If, however, the edge switch determines that there is a destination port conflict which prevents it from switching the burst, it refuses the the `setup` message and returns a `reject` message to the user. In this case, the user *drops* the burst, and it enters an idle period waiting for another burst. When the next burst arrives, the user repeats the above process by transmitting a new `setup` message to its edge switch.

The source 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, in which case the node is said to employ *estimated release*. Alternatively, the source may simply transmit a `release` message to indicate the end of its transmission. If this *explicit release* feature is implemented, the OBS nodes free the resources allocated to the burst upon receipt of the `release` message. Our study can take into account either

[1]We should note that the value of the offset field is not the same with the notation offset we defined in section 1.4.

explicit or estimated release, due to the inherent abstractions in the simulation model. The sequence of messages exchanged between a user and its edge switch is shown in Figure 3.1; if estimated release is used, then the `release` message at the bottom of Figure 3.1 is not transmitted.

Figure 3.2 shows the flow of signaling messages within the OBS network for a successful burst transmission. Once the edge switch decides to accept a `setup` message, it returns a `setup ack` message to the user, and it also forwards the `setup` message to the next OBS node in the path to the destination. Therefore, the `setup` message travels across the network informing the switches on its path of the burst arrival. If no blocking occurs along the path, the `setup` message eventually reaches the destination user; the data burst follows shortly thereafter. Upon receipt of the `setup` message, the destination user may optionally send a `connect` message to the originating user acknowledging the successful connection. The `connect` message travels along the same path as the `setup` message, but in the opposite direction.

If, upon receipt of a `setup` message, an intermediate switch determines that it cannot switch the incoming burst, it returns a `reject` message to the previous node (the one who sent the `setup` message), and it drops the burst. Upstream nodes receiving the `reject` message free the switch resources allocated to the burst, and forward the message along the reverse path back to the user transmitting the burst.

In our simulation, we explicitly model the transmission and processing of the `setup`, `setup ack`, `reject`, and `release` messages exchanged between users and edge switches, and between the OBS nodes, as well as the propagation delays involved. We do not model the optional `connect` message, since it does not affect the performance of the data plane of the network.

3.2 The Burst Traffic Model

As we discussed ealier, we assume that each user of the OBS network is connected to an edge switch using a bidirectional fiber that carries the same set of $W + 1$ burst wavelengths (one for transmitting signaling messages) in each direction. Each user is associated with a burst arrival process. In this study, we assume that the burst arrival process is a Poisson point process. A burst can be either long or short. The duration of a burst, whether

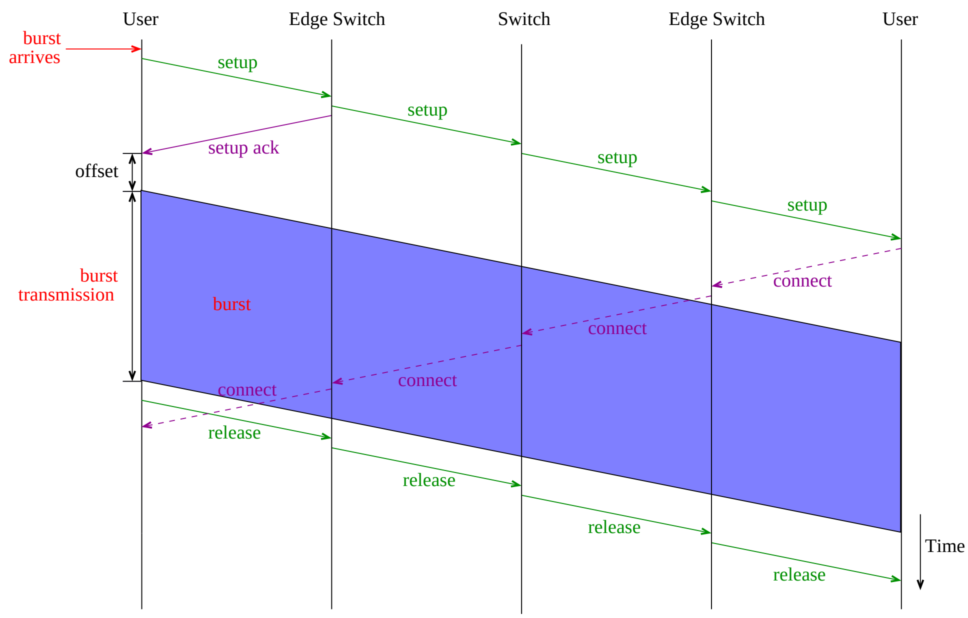

Figure 3.2: `JumpStart` JIT signaling messages in the OBS network for a successful burst transmission

short or long, is assumed to be exponentially distributed.

The burst arrival process of each user can be characterized completely by the following four parameters: **(1)** the parameter λ of the Poisson point process (note that $1/\lambda$ is the mean duration of the burst interarrival time); **(2)** the mean duration $1/\mu_l$ of a long burst; **(3)** the mean duration $1/\mu_s$ of a short burst; and **(4)** the probability p_s that a burst is a small burst.

Since a customer may request either a short or a long burst with probabilities p_s and $1 - p_s$, respectively, the burst size distribution is a two-stage hyper-exponential distribution, as shown in Figure 3.3. It is well-known that this distribution can be represented by a two-stage Coxian server [50]. As addressed in Chapter 2, the distribution of service time will not influence the performance of the network in terms of burst blocking probability.

In Chapter 2, we found the distribution of the burst length has no inluence on the burst blocking probability, and only the traffic intensity is important. However, in that chapter, we only considered the process of `setup` message. In this chapter, we also need to simulate other messages, particularly, the `reject` message. This message may cause an upstream switch to releases a burst before it completes its transmission, thus its actual service time is reduced. If the length of a burst is less than twice the switch-to-switch

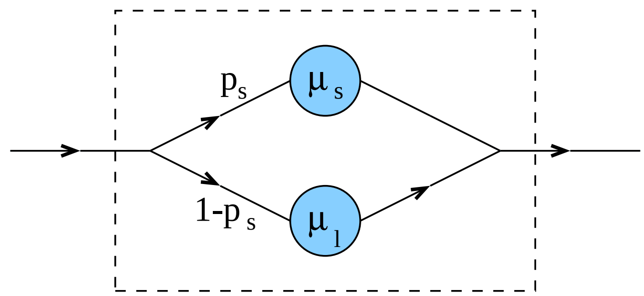

Figure 3.3: Hyperexponential server corresponding to the burst size distribution

signal propagation delay, when the `reject` message arrives at a switch to notify it that the burst has been dropped by the downstream switch, the last bit of the data of the burst has already left the switch. Thus the service time does not decrease. On the other hand, a very long burst is possible to be released by a `reject` message originated by a switch many hops away.

3.3 Simulations and Results

In this section we present simulation results of OBS networks with different topologies. The performance measure of interest is the *burst drop probability*, which is defined as the probability that a burst transmission will be unsuccessful, i.e., that a burst is dropped at some OBS node before it reaches its destination. We have used the method of batch means to estimate 95% confidence intervals for all the simulation results we collected. In our study, the number of batches was 30, with each batch run lasting until each switch has transmitted at least 120,000 bursts. Since, however, we have found that confidence intervals are very narrow, in order to improve the readability of the figures, we do not plot the confidence intervals.

In Table 3.1, we list the OBS network parameters we varied in this study. In all the simulations, we take the mean short burst duration $1/\mu_s$ as the unit of time (i.e., $1/\mu_s = 1$), and we express all other time quantities as multiples (or fractions) of this unit.

In this study, we assume that edge OBS nodes use the *random* wavelength assignment policy to select a free wavelength for users to transmit their bursts.

In the next subsection we consider a simple linear OBS network to gain insight into

Table 3.1: OBS network parameters

Parameter	**Description**
W	Number of burst wavelengths in a fiber link
C	Number of converters per OXC
$T_{setup}(JIT)$	Processing time of JIT messages at each OBS node
T_{OXC}	OXC configuration time at each OBS node
τ	Link propagation delay (one-way)
λ	Burst arrival rate per user (parameter of Poisson process)
$\frac{1}{\mu_s} = 1$	Mean duration of short bursts (the **unit** of time)
$\frac{1}{\mu_l}$	Mean duration of long bursts
p_s	Probability that a burst is a short one

how the various network parameters affect the burst drop probability. Then, in Section 3.3.2 we study networks of more general topologies.

3.3.1 A Linear OBS Network

Figure 3.4 shows the OBS network we study in this section. The network consists of four OBS nodes running the `JumpStart` JIT protocol. The nodes are connected in a unidirectional linear topology in which traffic flows from left to right only. Each OBS node serves exactly $2N$ users. Each user of the OBS network acts as either the source or the sink of burst traffic, but not both. Specifically, all $2N$ users attached to switch S_1 (the leftmost one in Figure 3.4) generate burst traffic, but they do not sink any traffic. On the other hand, all $2N$ users of switch S_4 (the rightmost one in Figure 3.4) only act as traffic sinks, they do not originate any bursts. Finally, exactly N of the users attached to each of switches S_2 and S_3 (in the middle of the linear network) generate but do not receive traffic, and exactly N users receive but do not generate traffic.

The traffic pattern in the linear network is as follows. The N users drawn to the left of switch S_1 generate bursts whose destination is one of the N users drawn to the right of switch S_4. Also, the N users drawn above switch S_i generate bursts whose destination is one of the N users drawn below switch $S_{i+1}, i = 1, 2, 3$. The traffic from switch S_1 to switch S_4 will be referred to as *through traffic*, while the traffic from switch S_i to switch S_{i+1} will be referred to as *cross traffic*. The objective of this study is to investigate the burst drop rate of through traffic under the presence of cross traffic, and how it is affected

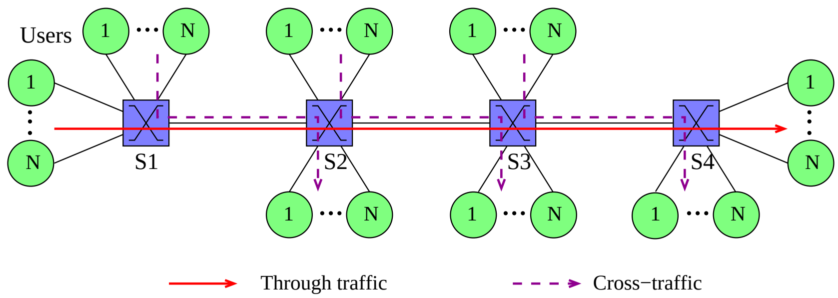

Figure 3.4: The linear OBS network

by the traffic load and the values of various system parameters, including the number of converters at each OBS node and the delays encountered at each switch (including message processing and OXC configuration).

As we explained above, we let the mean short burst duration $1/\mu_s = 1$ in our simulation, and we express all other time quantities in terms of this unit. In our simulation experiments, we varied the values of the various parameters as follows.

- Number of wavelengths in each fiber, $W = 128$.
- Number of users per switch, $2N = 32$.
- Number of converters per switch, $C = 0, \cdots, W$.
- User-to-switch delay is negligible.
- Switch-to-switch delay, $\tau_{switch} = 5, 10, 20, 40, 80, 160, 320$ units.
- OXC fabric configuration time, $T_{OXC} = 0.01, 0.1, 1$ units.
- JIT message processing time at each OBS node, $T_{JIT} = 0.01, 0.1, 1\ T_{OXC}$.
- Burst interarrival time at each user, $1/\lambda = 10$ units.
- Mean long burst size, $1/\mu_l = 100$ units.
- Probability of short burst, $p_s = 0.8$.

The Effect of Wavelength Conversion

In this section we investigate how the degree of wavelength conversion at the OBS nodes affects the burst drop probability. We use a large number of wavelengths to make the reason for burst loss is mainly from the lack of wavelength converters instead of the lack of wavelengths. And we use a fixed switch-to-switch propagation delay $\tau_{switch} = 20$ units.

Figure 3.5 plots the burst drop probability for both through traffic and cross traffic in the linear network when the burst arrival rate of each user (that can generate bursts) is $\lambda = 0.15$. For the through traffic, we show four different drop probabilities: the *overall* drop probability, defined as the probability that a burst generated by the users of switch S_1 is dropped *anywhere* in the path to its destination at switch S_4, and the three *switch* drop probabilities at switches S_1, S_2, and S_3, defined as the probability that a burst *arriving* at switch $S_i, i = 1, 2, 3$, respectively, will be dropped by the switch. It is easy to see from Figure 3.4 that no bursts are ever dropped at switch S_4, therefore its switch drop probability is zero. We also note that the arrival rate of through traffic bursts is different at the various switches, since some bursts are dropped at upstream switches. Therefore, adding the three switch drop probabilities *does not* yield the overall drop probability for the through traffic.

We make several observations from Figure 3.5. First, we see that when no conversion is available (i.e., the number of converters $C = 0$ in the figure), the overall drop probability of through traffic, as well as the drop probability of through traffic at switches S_2 and S_3 is much higher (by four orders of magnitude) than the drop probability of through (and cross) traffic at switch S_1. We can even find the burst blocking probabilities of cross traffic in switch S_2 and S_3 are both zero. This result can be explained as follows. Recall that when it receives a `setup` message from a user, the edge switch checks for a free wavelength at the desired output port. If no such wavelength exists, and since no conversion is available, the switch sends a `reject` message to the user. If, however, one or more free wavelengths are found, the edge switch selects one, say, wavelength w, using the random wavelength assignment policy [61]. It then includes this wavelength in the `setup ack` message it returns to the user, effectively instructing the user to transmit the burst on wavelength w; it also reserves wavelength w on this output port so it cannot be used by other users. Since the wavelength on which a user transmits is selected by the edge switch in this manner, a user's `setup` message is rejected if and only if *all* wavelengths on the desired output port are busy transmitting other bursts. Since cross traffic bursts are 1-hop, so virtually they may be dropped at the edge switch only, it is expected that their drop probability will be low, as the results in Figure 3.5 confirm. Similarly, the drop probability of through traffic at its edge switch S_1 is also low.

In fact, according to our previous analysis, the burst drop probabilities of cross traffic and through traffic at S_1, are decided (or approximated) by the `Erlang's blocking formula`. While the number of wavelengths is very large, i.e. 128, their drop probability will become very small. Furthmore, since the overall input traffic to switch S_1 is heavier than those to switch S_2 and S_3, the burst drop propability of the cross traffic at the latter switches are even smaller (going to zero).

On the other hand, since no converters are available, a through traffic burst can only be switched by downstream nodes if and only if the *same* wavelength w is available on the desired output port. Since this wavelength w may already be in use by a cross traffic burst at the time the `setup` message for the through traffic burst arrives, the latter may be rejected and the corresponding burst dropped. As a result, through traffic bursts experience very high drop probability at switches S_2 and S_3 (as well as overall), while cross traffic bursts experience very low drop probability at the same switches. In fact, because of the very low drop rate of cross traffic bursts, more of them are admitted in the network, causing many through traffic bursts to be dropped, which in turn allows more cross traffic bursts to be successful, and so on, driving the drop probability of through traffic to very high levels.

The second observation we make from Figure 3.5 is that if converters are available at each OBS node, the drop probability for through traffic (both overall and at switches S_2 and S_3) is dramatically lower than with no converters, while the drop probability for cross traffic (and for through traffic at switch S_1) is higher. To explain this result, note that when employing converters, an intermediate switch has greater flexibility in accommodating through traffic bursts: a burst arriving on wavelength w can be switched as long as wavelength w is free on the desired outgoing port or if there is at least one available converter to convert the wavelength to another one that is free on the output port. In essence, each intermediate switch can now select the outgoing wavelength of a burst, similar to the manner in which the edge switch selects the wavelength on which a user initially transmits a burst. As a result, compared with the no-converter case, fewer through traffic bursts are dropped at intermediate nodes, significantly decreasing the drop probability of such bursts. Not dropping these through traffic bursts means that the output ports of all switches remain occupied by these bursts for longer periods of time, therefore, the drop probability for cross traffic bursts tends to increase with the number of converters.

From Figure 3.5 we also see the law of diminishing returns in action. As we increase the number of converters from zero to 4, the drop probability of through traffic at switch S_2 decreases dramatically, but additional converters have a smaller effect[2]. Actually it sometimes increases. The reason for this is that when the number of wavelength converters increases, the burst drop probability of through traffic in switch S_3 decreases. Let us recall when S_3 decides to block a through burst, it will send back a `reject` message to S_2, and the latter will terminate the service of the burst (though it has already accepted this burst) to enhance efficiency[3]. So, when burst drop probability at S_3 decreases, S_2 has less chance to "reduce" the service time of bursts and this will incur a higher utilization and hence cause a higher burst blocking probability. This reason can also explain that when number of wavelength converters increases, the through (and cross) traffic burst blocking probability at S_1 increases. And we expect, when the switch to switch delay increases, the increase in burst blocking probability will become less obvious. And this will be shown in the following subsection.

We can find a "strange" observation that when the wavelength converters are used, the through traffic burst blocking probability at switch S_2 is less that at switch S_3. It is counter-intuitive since the overall traffic input to S_2 and going out through the link S_2S_3, which is the aggregation of through traffic coming from S_1 and cross traffic from users of S_2, should be larger than the overall traffic input to S_3 and going out through link S_3S_4, which is the aggregation of through traffic coming from S_2 and cross traffic from users of S_3. The loads of the cross traffic components of these two traffics are similar, however the through traffic output from S_1 should be larger than that from S_2. This means that S_3 has less traffic to link S_3S_4 than traffic at S_2 to link S_2S_3, while it has a larger through burst blocking probability. And this becomes more confusing when we observe that the cross traffic burst blocking probability of S_3 is less than that of S_2. We also ascribe the results to the `reject` message, and will explain it in detail in the next subsection.

Figures 3.6-3.8 are similar to Figure 3.5, but correspond to lower burst arrival rate, namely 0.1, 0.08, and 0.05, respectively. We should note that all the cross traffic has zero burst blocking probability in our simulation, and the burst blocking probability of

[2] S_3 also has such a critical point except the number of converters is 60.

[3] In JIT, service time includes offset and transmission.

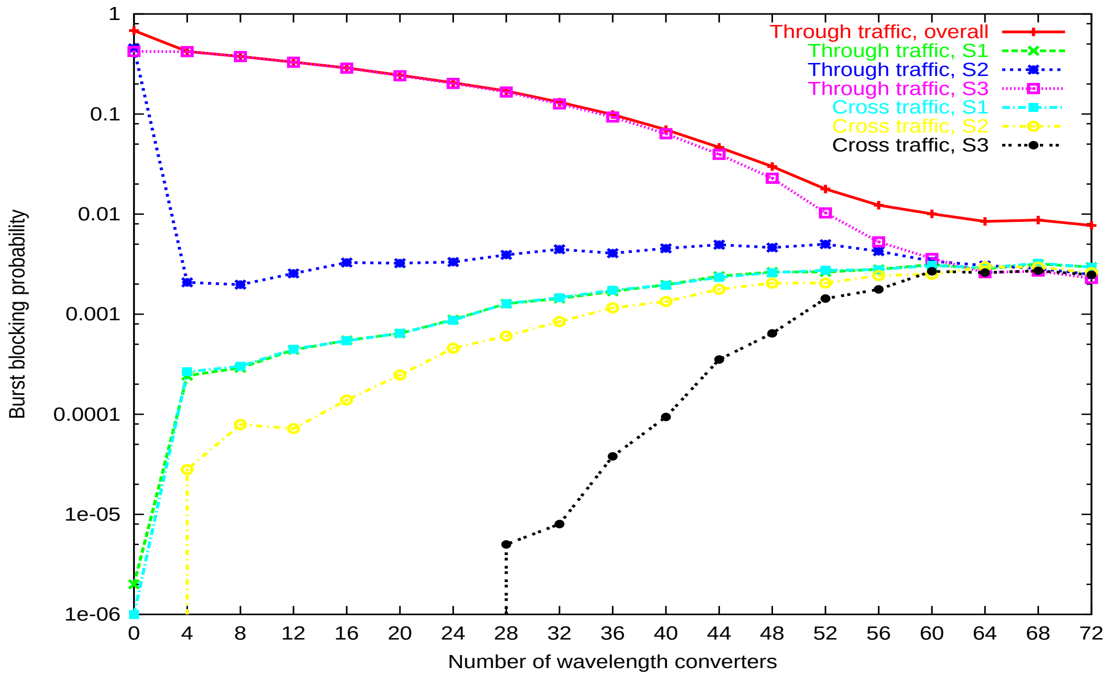

Figure 3.5: Burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.15$

through traffic in switch S_1 is also zero. So they are not shown in the figures for the sake of readability.

As we can see, as the burst arrival rate decreases from 0.15 to 0.05, all burst drop probabilities decrease. However, the relative behavior of the various curves as the number of converters increases is very similar to that that in Figure 3.5, and can be explained in a similar manner.

The Effect of Switch to Switch Delay

In our previous experiments, we assume that switch-to-switch propagation delay τ_{switch} does not have any significant effect on the burst drop probability in the network. This is justified by the fact that, since both the JIT signaling messages and the data bursts propagate at the same speed of light and traverse the same links, the propagation delay does not affect the amount of time that bursts occupy network resources. Therefore, throughout the previous study, we use the value $\tau_{switch} = 20$ in our simulations.

However, in the `Jumpstart` protocol, when a switch can not accept a burst, it will

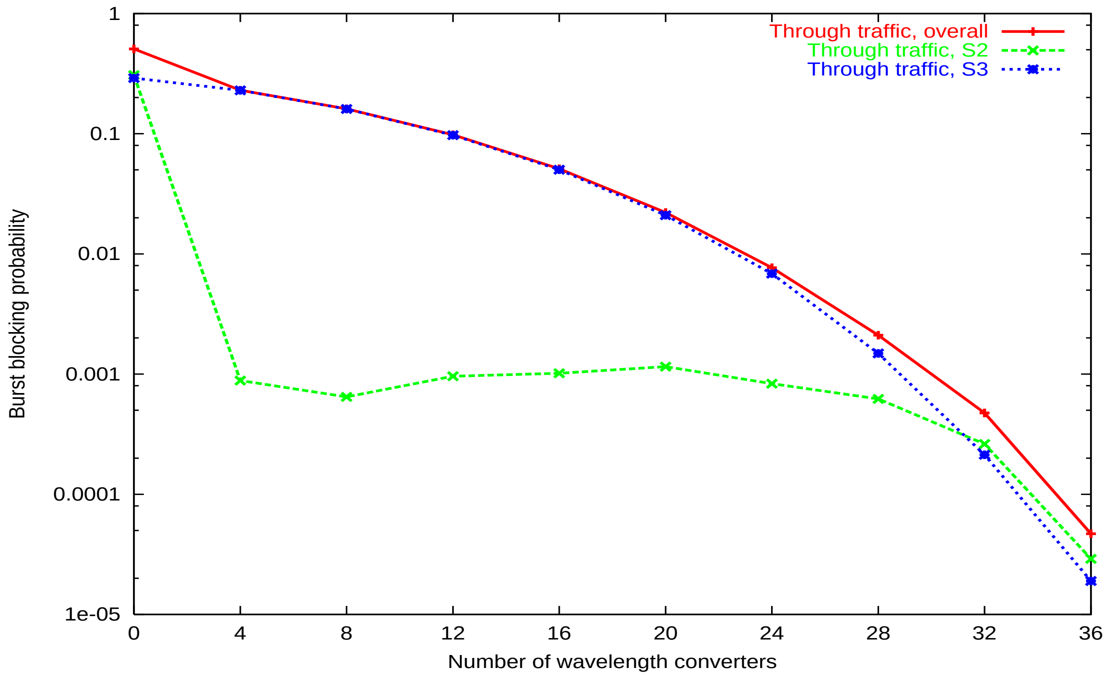

Figure 3.6: Burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.1$

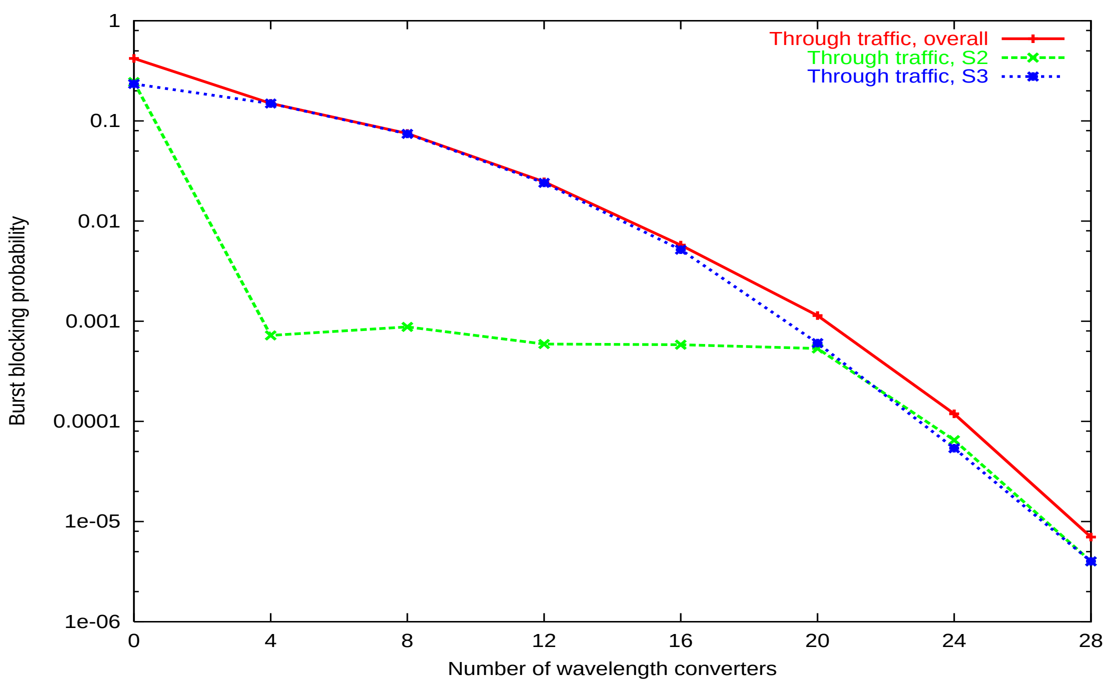

Figure 3.7: Burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.08$

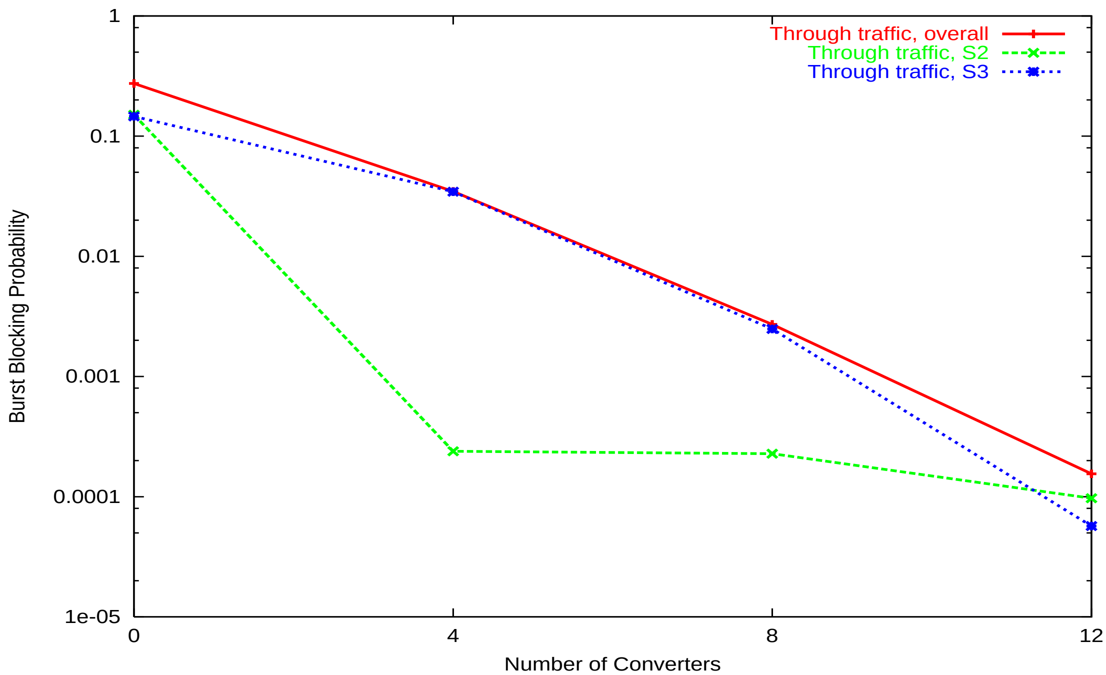

Figure 3.8: Burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.05$

send back a `reject` message to its upstream switch. So the latter will release the resource it assigned to that burst to enhance the efficiency.

As we mentioned in Section 3.3.1, in Figure 3.5 the through traffic burst blocking probability at S_2 is lower than that at S_3 when the number of wavelength converters is not large enough. We believe that the main reason for this phenominon is the `reject` messages. In Figure 3.9, we show the results using the same values for the system parameters with those of Figure 3.5 except that the switch to switch delay is 320 units. This delay is rather long so that when a downstream switch decide to reject a burst, and it sends back a `reject` message to its upstream switch, due to the long switch to switch delay, this message will arrive at the upstream switch after the burst has already been transmitted by that switch. This will make `reject` messages help little in performance. In Figure 3.9, we find the through traffic burst blocking probability at S_2 is always higher than that at S_3. As a comparison, in Figure 3.10, we show the results in the situation that no `reject` message is used, and we find these results are similar to those of Figure 3.9, so our proposition is justified.

Furthermore, in Figures 3.11-3.13, we show the overall through traffic burst block-

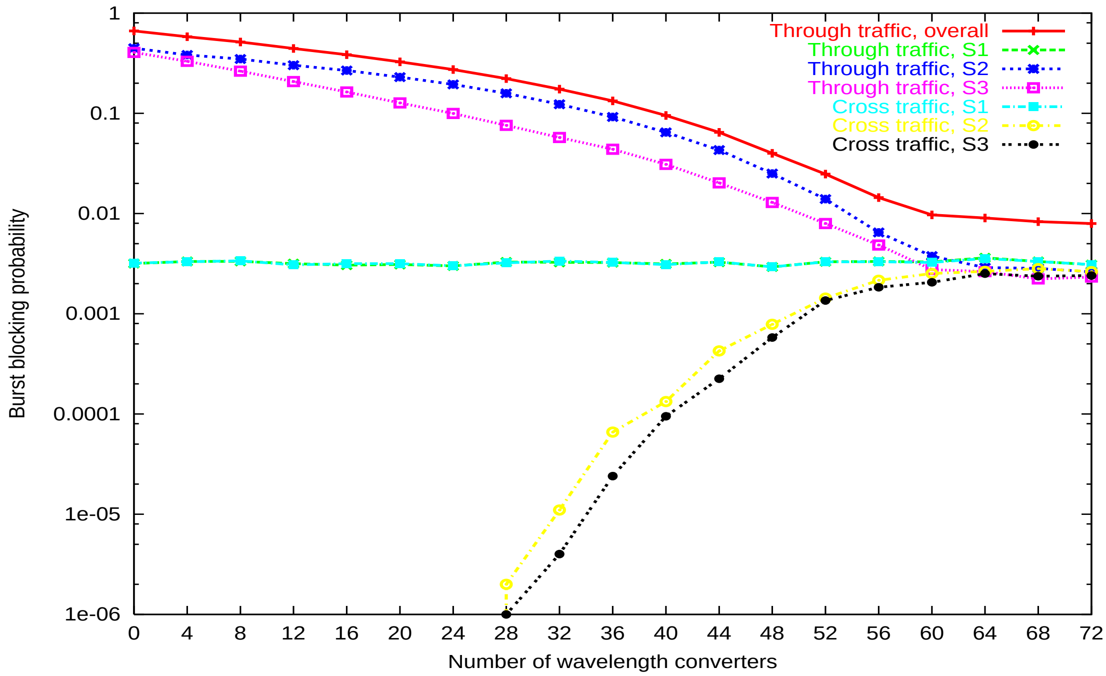

Figure 3.9: Burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.15$,$\tau_{switch} = 320$

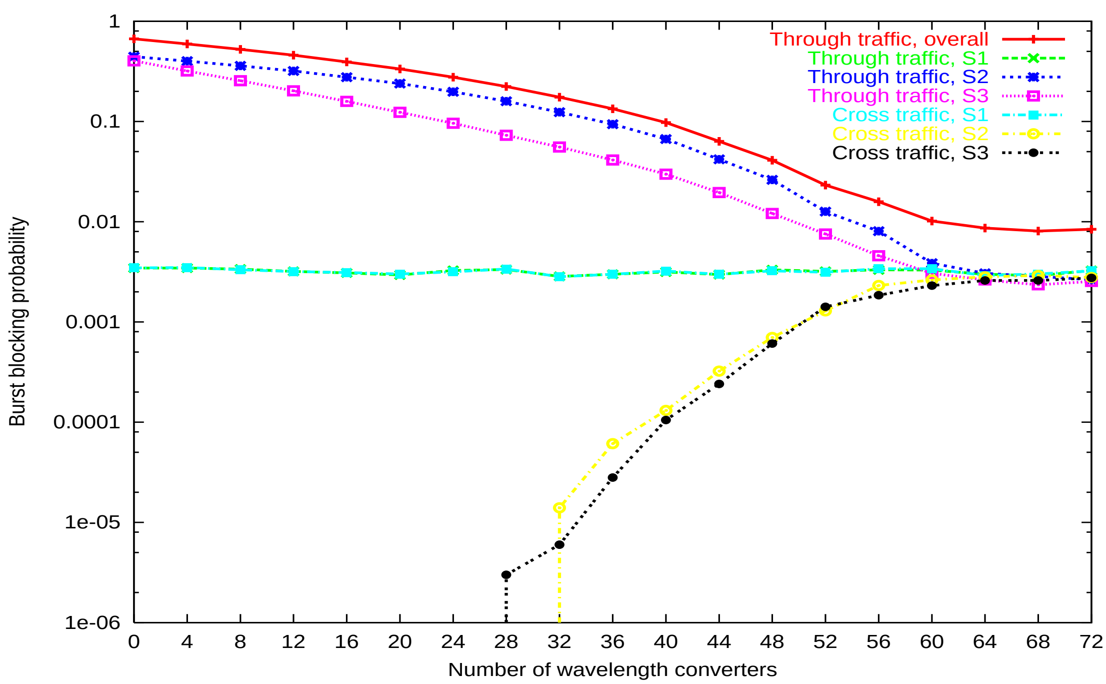

Figure 3.10: Burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.15$

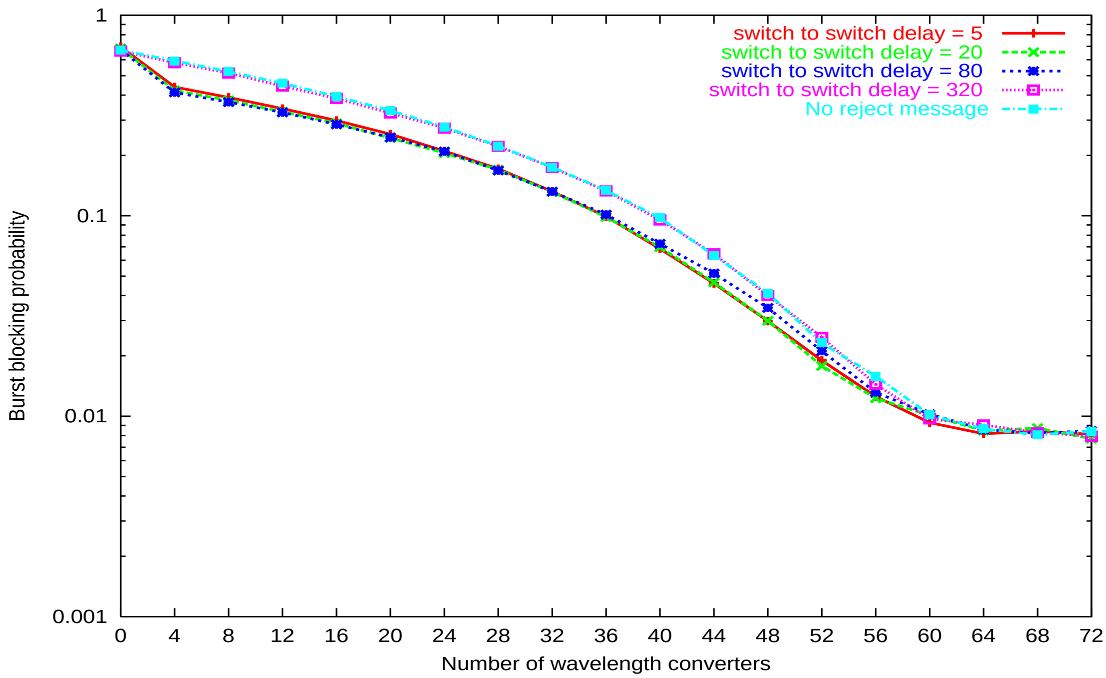

Figure 3.11: Overall through traffic burst blocking probability, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.15$

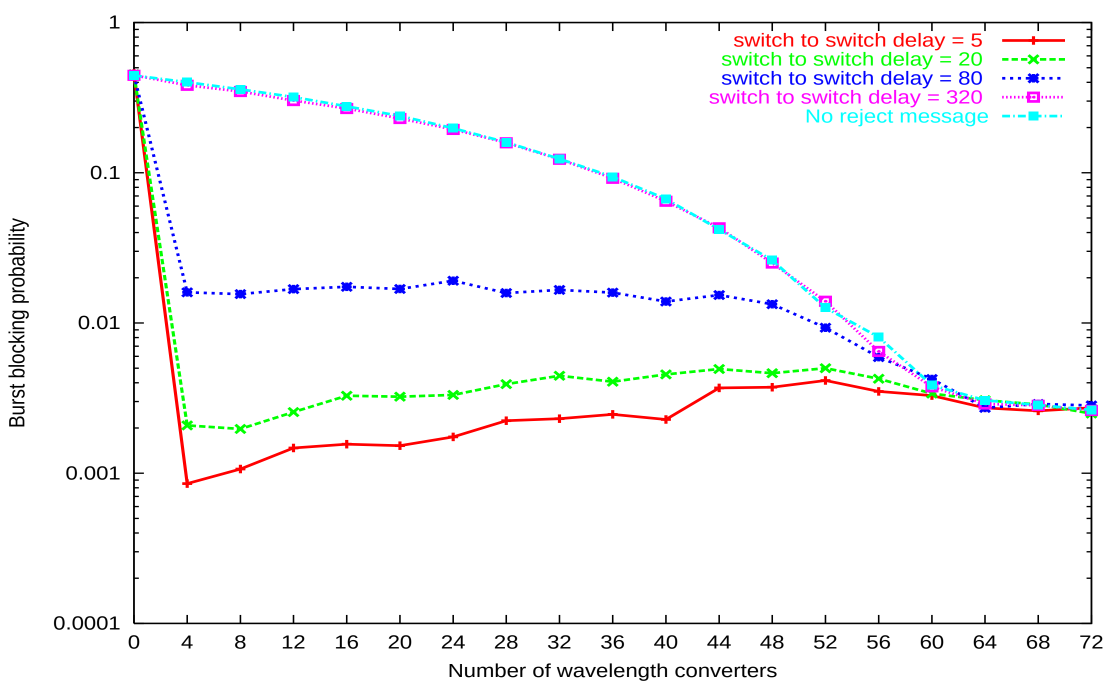

Figure 3.12: Through traffic burst blocking probability at switch S_2, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.15$

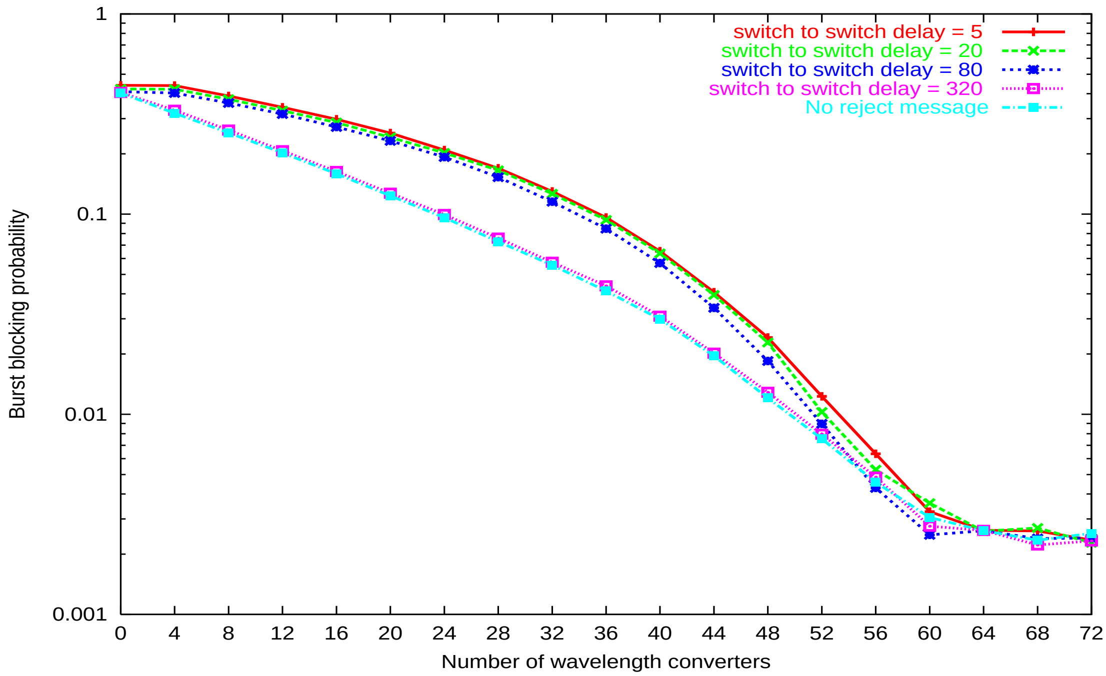

Figure 3.13: Through traffic burst blocking probability at switch S_3, $T_{setup}(JIT) = 0.01$,$T_{OXC} = 1$,$\lambda = 0.15$

ing probability and through traffic burst blocking probability at switch S_2 and S_3, respectively, under the variance of switch to switch delay. As we find, although the blocking probability at S_2 increases dramatically with the increase of τ_{switch} (when the number of wavelength converters is not big enough), the burst blocking probability at S_3 decreases continuously, thus the overall blocking probability changes little. And we also observe that the overall burst blocking probability in different τ_{switch} are close to the upper bound, where no `reject` message is used. So we may expect that the variance of the switch to swich delay does not have great influence on the overall performance since its influence on one node may be canceled by its influence on another node.

The Effect of Message Processing and OXC Configuration Delays

We now investigate the effect of the message processing delay $T_{setup}(JIT)$ and the OXC configuration delay T_{OXC} on the burst drop probability of through traffic. Recall from expression (1.1), that these two parameters determine the value of the burst offset. As we described earlier, the offset is included in service time in JIT. Thus we expect that large offset values (i.e., large values of $T_{setup}(JIT)$ and T_{OXC}) will have a negative effect on burst drop probability, since switches must reserve resources for a large period of time relative to the burst size.

In Figures 3.14 and 3.15 we plot the burst drop probability of through traffic in the linear network shown in Figure 3.4, against the values of $T_{setup}(JIT)$ and T_{OXC}. For simplicity, we let $T_{setup}(JIT) = 0.01T_{OXC}$, and we vary the value of T_{OXC} from 0.01 units to one unit of time. We show three different curves in each figure, each corresponding to a different burst arrival rate of one user. Figure 3.14 shows results for $C = 8$ converters per switch, while Figure 3.15 shows results for $C = 12$ converters.

The results in Figures 3.14 and 3.15 confirm the above observations. As we can see, the values of the drop probabilities increase as the value of T_{OXC} increases from 0.01 to 1, reflecting the larger amount of time that resources are reserved for each burst (and thus, are unavailable to other bursts) due to larger offset values. However, as T_{OXC} increases from 0.01 to 0.1, there is only a slight increase in burst drop probability. This result is due to the fact that for such small values of T_{OXC}, the offset is only a small fraction of the average burst size (recall that the unit of time is equal to the mean duration of short bursts). Therefore, the additional amount of time that resources are reserved (wasted) at each switch is small relative to the actual burst size. However, as T_{OXC} increases to 1 or beyond, the offset value becomes comparable to the average burst size, and thus, resources are reserved for amounts of time much larger than the burst length, leading to inefficient utilization of network resources and poor performance. We observe that this behavior is consistent across a wide range of load values, as well as for different degrees of conversion.

Overall, the results in Figures 3.14 and 3.15 suggest that the `Jumpstart` JIT protocol has good performance as long as the values of the message processing delay, $T_{setup}(JIT)$, and the OXC configuration delay, T_{OXC}, are kept small compared to the average burst size.

3.3.2 OBS Networks of General Topologies

In this section, we present the simulation results for two 16-node OBS networks. The nodes of the first network are arranged in a regular topology, namely, a 4×4 torus, while the second network has the topology of the NSFNet. In this study we consider shortest path routing. When an OBS node receives a `setup` message, it checks to see if it can forward the corresponding burst over the output port on the shortest path to the destination. Here,

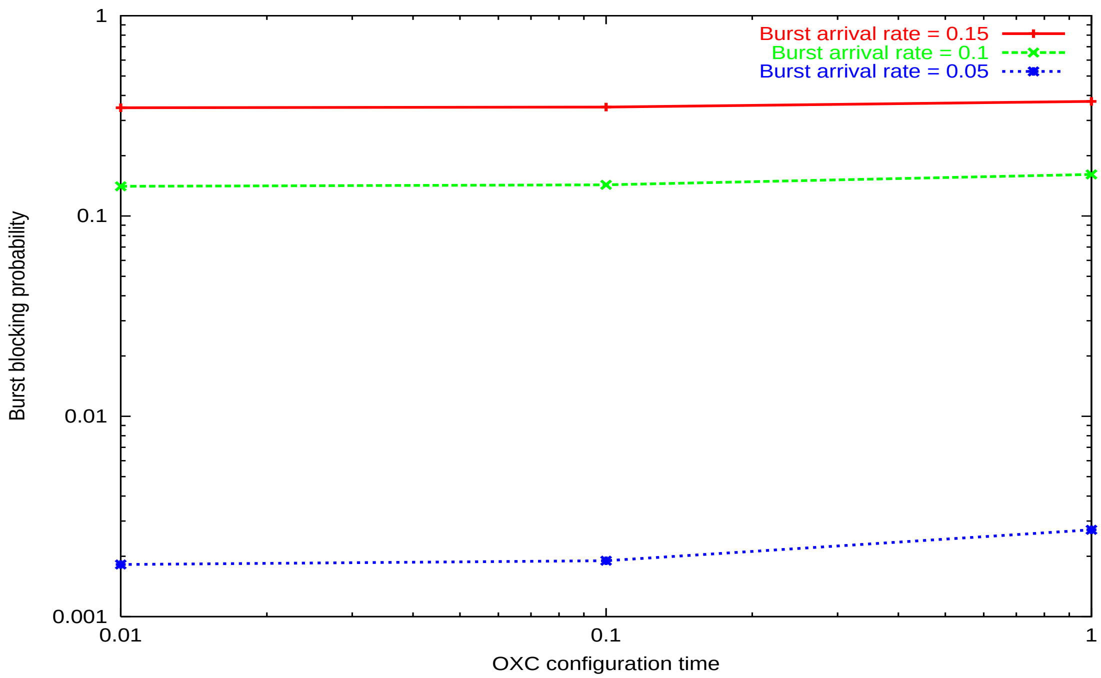

Figure 3.14: Overall through traffic burst drop probability, $C = 8$ converters, $T_{setup}(JIT) = 0.01T_{OXC}$

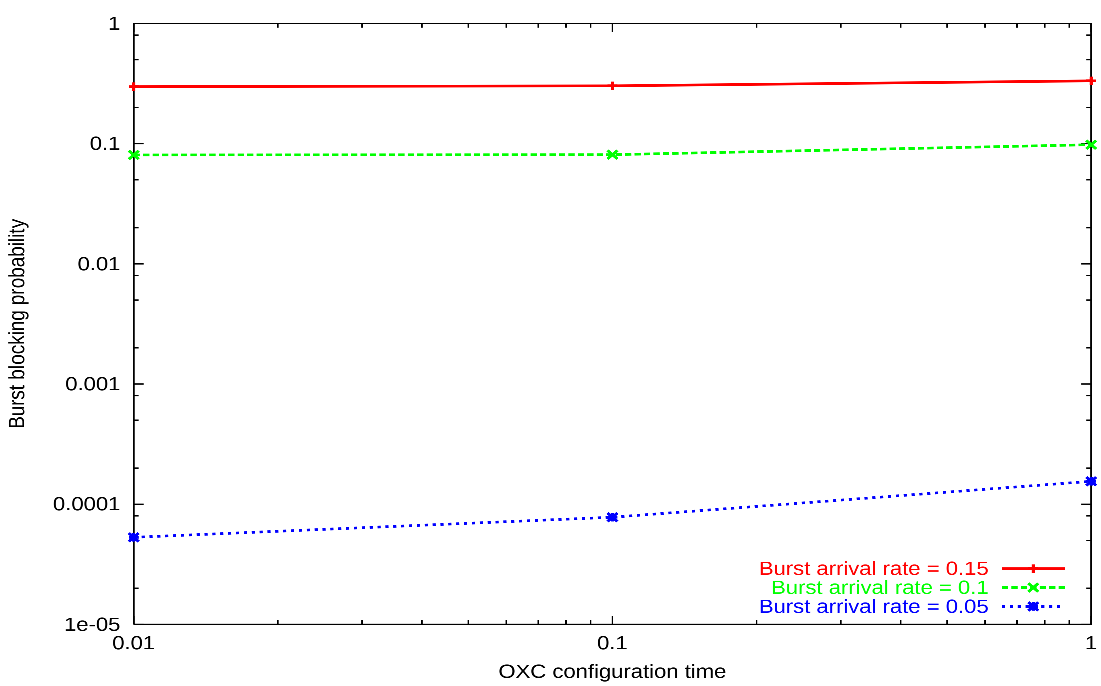

Figure 3.15: Overall through traffic burst drop probability, $C = 12$ converters, $T_{setup}(JIT) = 0.01T_{OXC}$

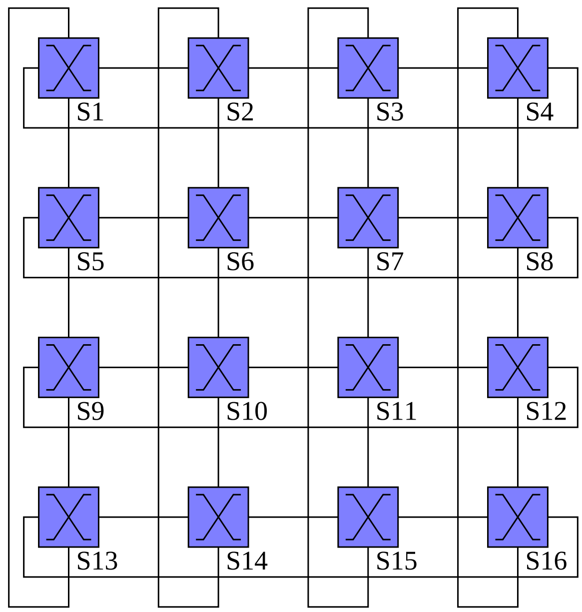

Figure 3.16: The 4×4 torus network

"shortest path" refers to the one with the minimum overall delay; if there are multiple such paths, one is selected arbitrarily and is used for all communication between the given pair of nodes. If the selected shortest path is available, then the node proceeds with the burst transmission. Otherwise, it drops the burst without considering alternate paths to reach the destination. In this simulation, we set the number of users 32 and the number of wavelengths 128, and we use the *random* wavelength assignment policy [61].

The 4×4 Torus Network

We consider the 4×4 torus network shown in Figure 3.16. Each switch is connected to four neighbors using bidirectional fibers, each capable of carrying W burst wavelengths (plus the control wavelength). We also assume that each switch serves a number of users, and acts as an edge node for these users. The traffic pattern we used is a symmetric one, and is such that: **(i)** each switch has the same number of users attached to it; **(ii)** each user of each switch generates burst at the same rate; and **(iii)** the destination switch of a burst generated by a user attached to switch i is equally likely to be any of the fifteen switches other than i.

In Figures 3.17, we plot the drop probability of all the bursts originating at users in this network, against the number C converters at each switch. (Recall that the wavelength converters are *shared* among all output ports of a switch, as shown in Figure 1.3.) For the results in Figure 3.17, the burst arrival rate at each user was set to 0.15. For these simulation experiments, we let the OXC configuration delay, T_{OXC} be 1 unit, the message processing delay $T_{setup}(JIT)$ be $0.01T_{OXC}$. The switch-to-switch propagation delay is 20 units, and the user-to-switch propagation delay τ_{user} is neglected. We note that, for the 4×4 Torus network shown in Figure 3.16, the shortest path between any two nodes can have 1-4 links. For instance, consider switch $S1$. Its immediate (one-hop) neighbors are switches $S2$, $S4$, $S5$, and $S13$; its two-hop neighbors are $S3$, $S6$, $S8$, $S9$, and $S14$; its three-hop neighbors are $S7$, $S10$, $S12$, and $S15$; and its four-hop neighbors are $S11$ and $S16$. Therefore, in Figures 3.17 we present four curves, each curve corresponding to the drop probability of bursts with a path length equal to one of the four possible values, one through four.

From Figure 3.17, we observe that the burst drop probability strongly depends on the length of the path that the burst has to travel. More specifically, the shorter the path (according to the number of hops), the lower the probability that the burst will be dropped at some switch along the path. While this result is expected, we note from the figure that the difference between the drop probability of bursts whose shortest path is one hop and that of bursts whose shortest path is four hops can be more than one order of magnitude.

We also see a dramatic decrease in burst drop probability as the number of converters increases, consistent with our observations of the linear network in the previous section.

Figure 3.18 is similar to Figure 3.17, except that burst arrival rate of each user is lower (0.1 instead of 0.15). As a result, the burst drop probabilities are lower than the corresponding ones of Figure 3.18. However, the relative behavior of the curves in the two figures is very similar.

Finally, in Figure 3.19 we show the effect of the message processing delay $T_{setup}(JIT)$ and the OXC configuration delay T_{OXC} on the drop probability of bursts whose paths are four hops long. For simplicity, we let $T_{setup}(JIT) = 0.01T_{OXC}$, as previously, and we plot the burst drop probability as the value of T_{OXC} varies from 0.01 units to one unit of time. We show two different curves in the figure, one for a burst arrival rate per user 0.15, and one for a rate of 0.1. The number C of converters in this experiment is set to four.

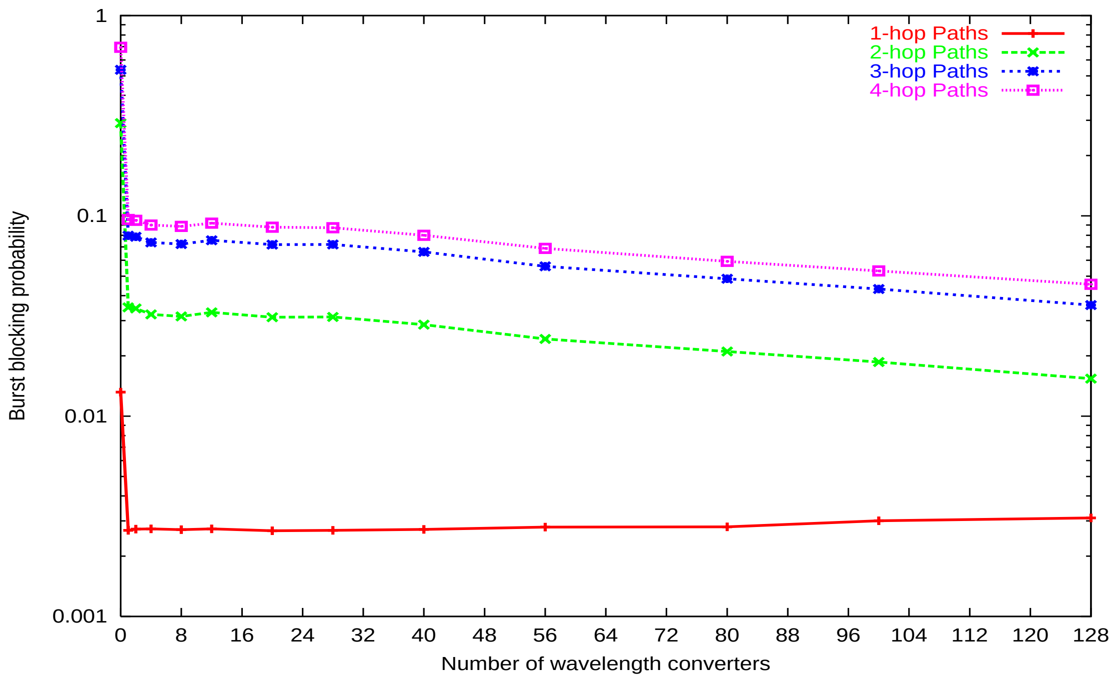

Figure 3.17: $T_{setup}(JIT) = 0.01T_{OXC}$, $T_{OXC} = 1$, $\lambda = 0.15$

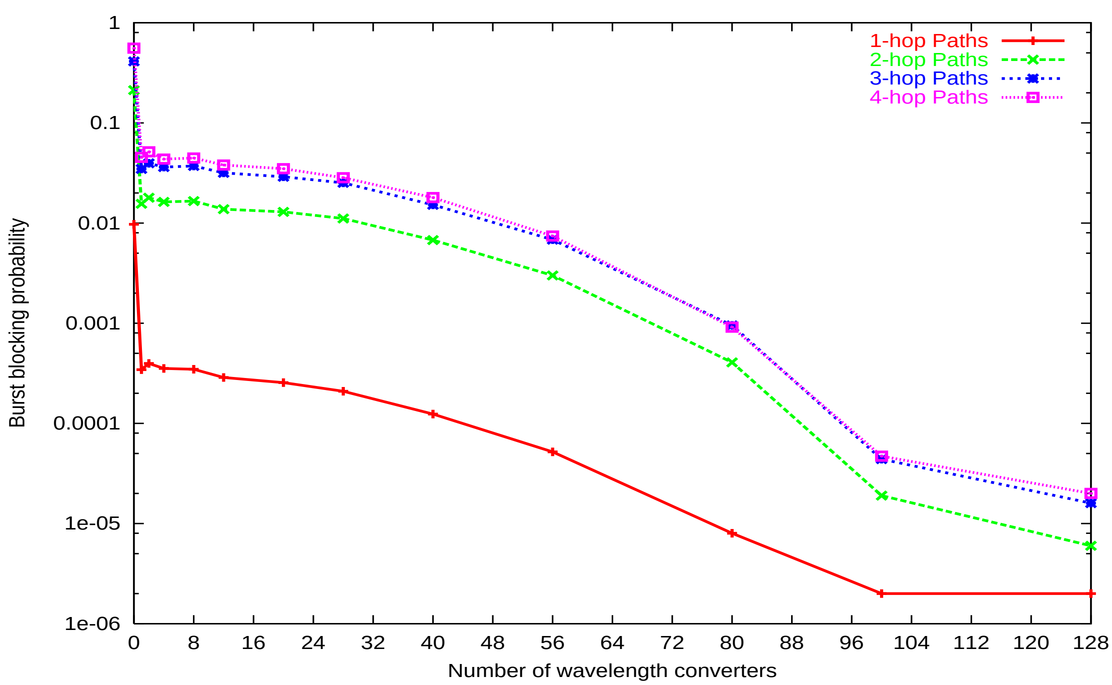

Figure 3.18: $T_{setup}(JIT) = 0.01T_{OXC}$, $T_{OXC} = 1$, $\lambda = 0.1$

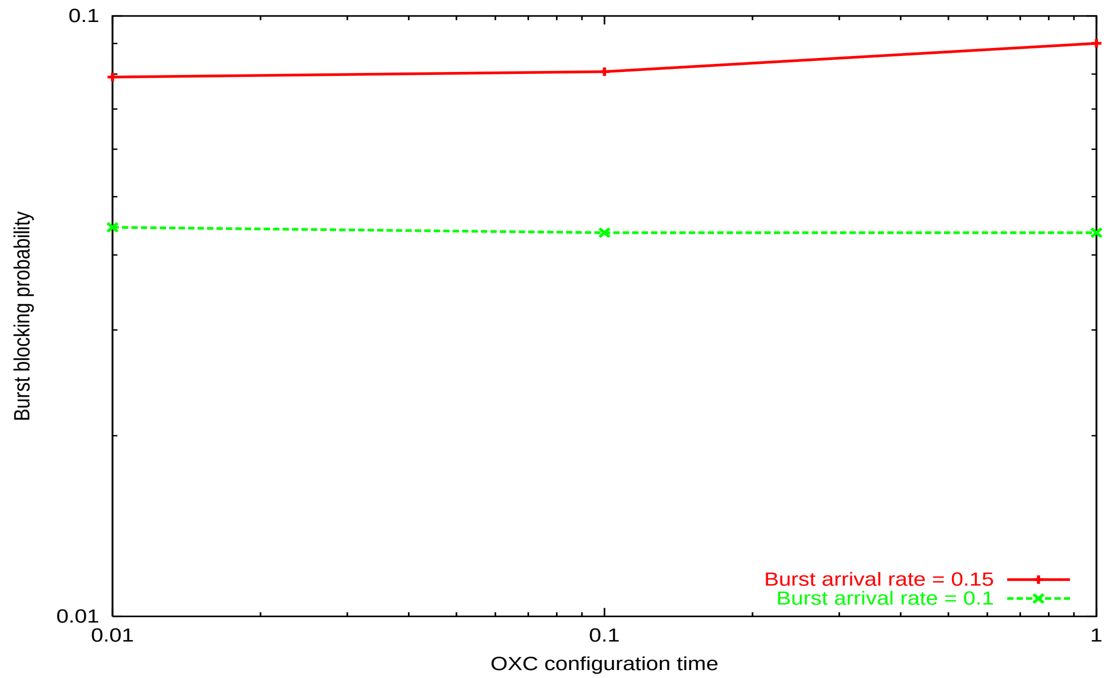

Figure 3.19: Burst drop probability of 4-hop paths, $C = 4$ converters

As we can see, increasing the value of T_{OXC} results in an increase in the burst drop probability. This increase reflects the larger amount of time that resources are reserved for each burst due to the larger offset values. The increase in burst drop probability is small as T_{OXC} increase from 0.01 to 0.1 units; for these small values, the offset length is small compared to the burst length. However, as T_{OXC} increase to one unit and beyond, the offset length becomes comparable to the burst length, and the burst drop probability increases accordingly. This behavior is consistent to that of the linear network (refer to Section 3.3.1.

The NSFNet

We now present simulation results for the irregular topology shown in Figure 3.21. This 16-node topology was obtained by augmenting the 14-node NSFNet topology through the addition of two fictitious switches, switch $S1$ and switch $S16$ in Figure 3.21, to capture the effect of NSFNet's connections to Canada's communication network, CA*net. In the resulting topology, the maximum length of a shortest path between any pair of nodes is equal to five hops. In simulation, the unit of time is chosen as $100\mu s$.

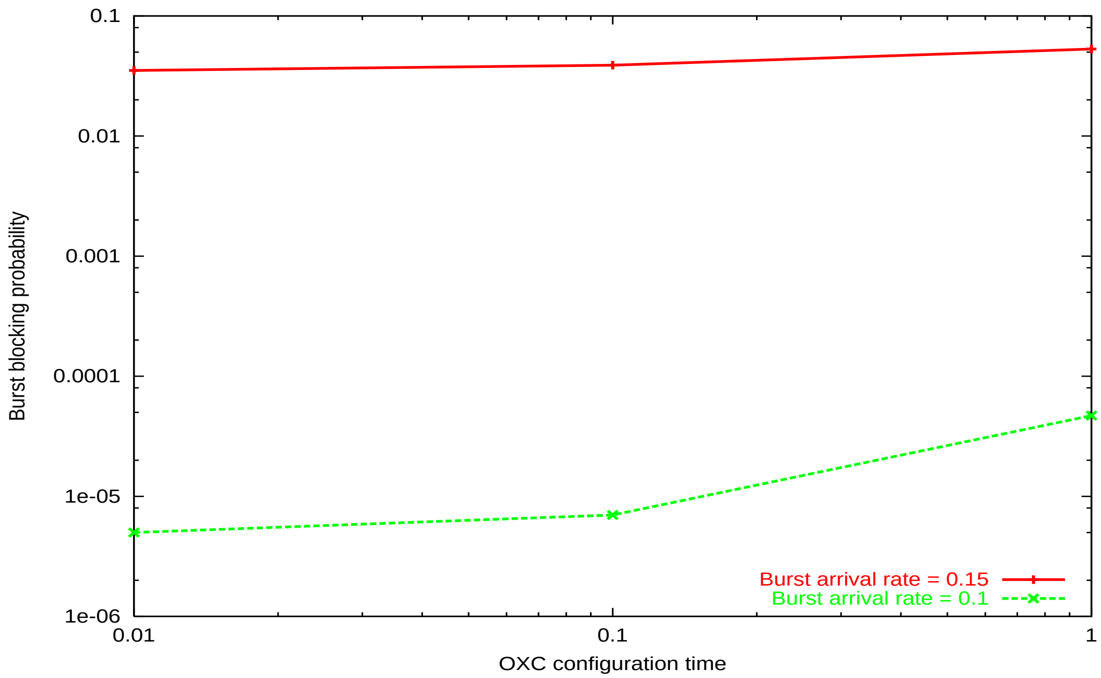

Figure 3.20: Burst drop probability of 4-hop paths, $C = 100$ converters

Figure 3.22 plots the drop probability of bursts against the number C of converters at each switch. Each of the five curves in the figure corresponds to the drop probability of bursts with a path length equal to one of five possible values, one through five. The behavior of the five curves is very similar to that of Figure 3.17 for the Torus network. In particular, we observe the dependence of drop probability to the length of a burst's path, as well as a similar decrease in drop probability as the number of converters per switch increases.

Finally, Figure 3.23 illustrates the effect of $T_{setup}(JIT)$, and T_{OXC} on the drop probability of bursts whose paths are five hops long, when the number of converters is $C = 4$. As in Figure 3.19, we let $T_{setup}(JIT) = 0.01T_{OXC}$, and we let the value of T_{OXC} vary from 0.01 units to one unit of time. As before, we observe that an increase in T_{OXC} results in an increase in the burst drop probability. This increase is small when T_{OXC} increases from 0.01 to 0.1 units. However, for values of T_{OXC} equal to or larger than one unit, the increase in burst drop probability becomes significant, as expected.

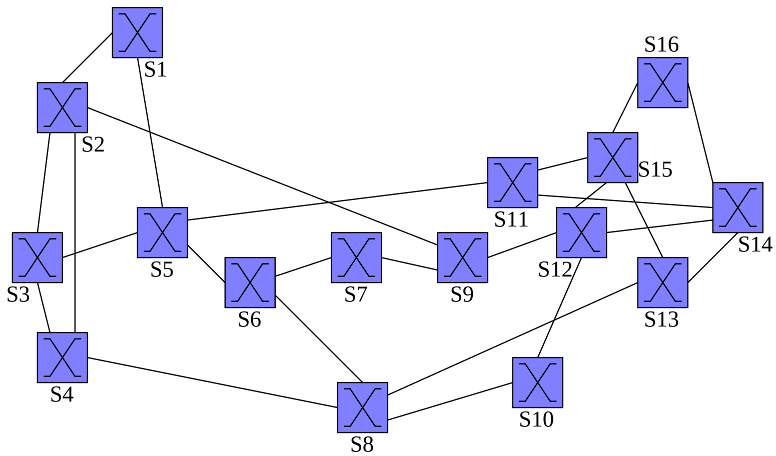

Figure 3.21: The 16-node topology based on the 14-node NSFNet

3.4 Conclusions

We have developed a simulation model of an optical burst switching network running the `Jumpstart` Just-In-Time signaling protocol. We have used the model to study the performance of various network topologies in terms of burst drop probability, and to investigate the effects of several system parameters, including message processing time, OXC configuration delay, and propagation delay.

We found that when the message processing time and the OXC configuration delay are small, they do not have great influence on the performance of the network in terms of burst blocking probability. We also found the switch-to-switch delay does influence the performance, however, its influence on different nodes of the network tends to cancel each other. Finally the number of wavelength converters deployed at each node is an important factor.

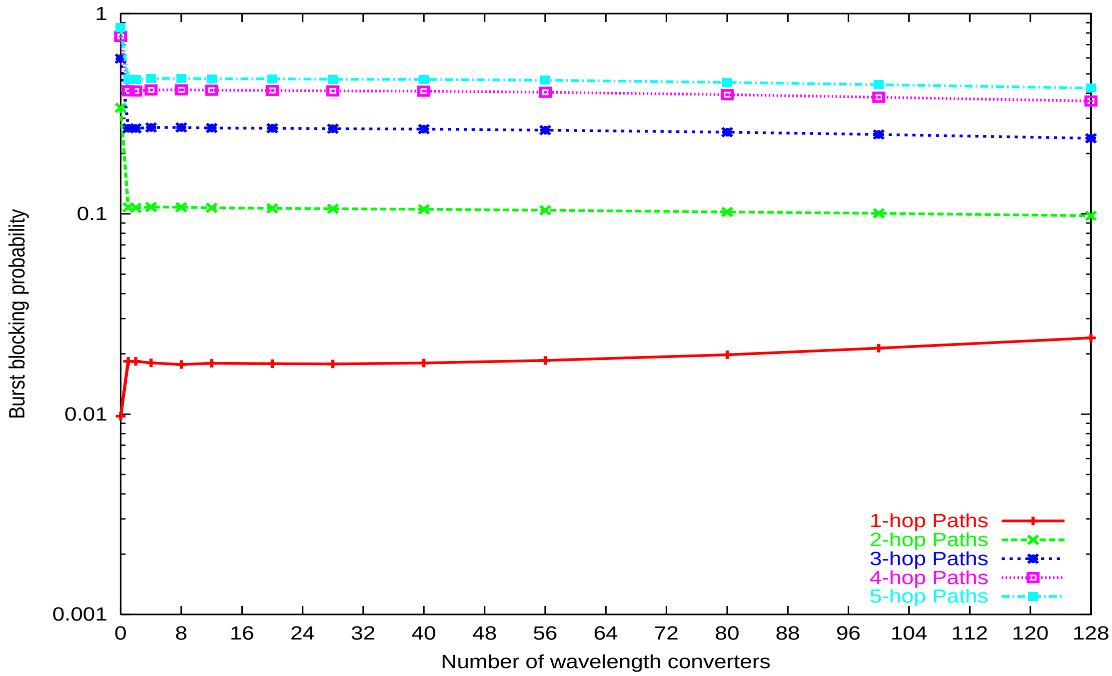

Figure 3.22: $T_{setup}(JIT) = 0.01T_{OXC}$, $T_{OXC} = 1$, $\lambda = 0.15$

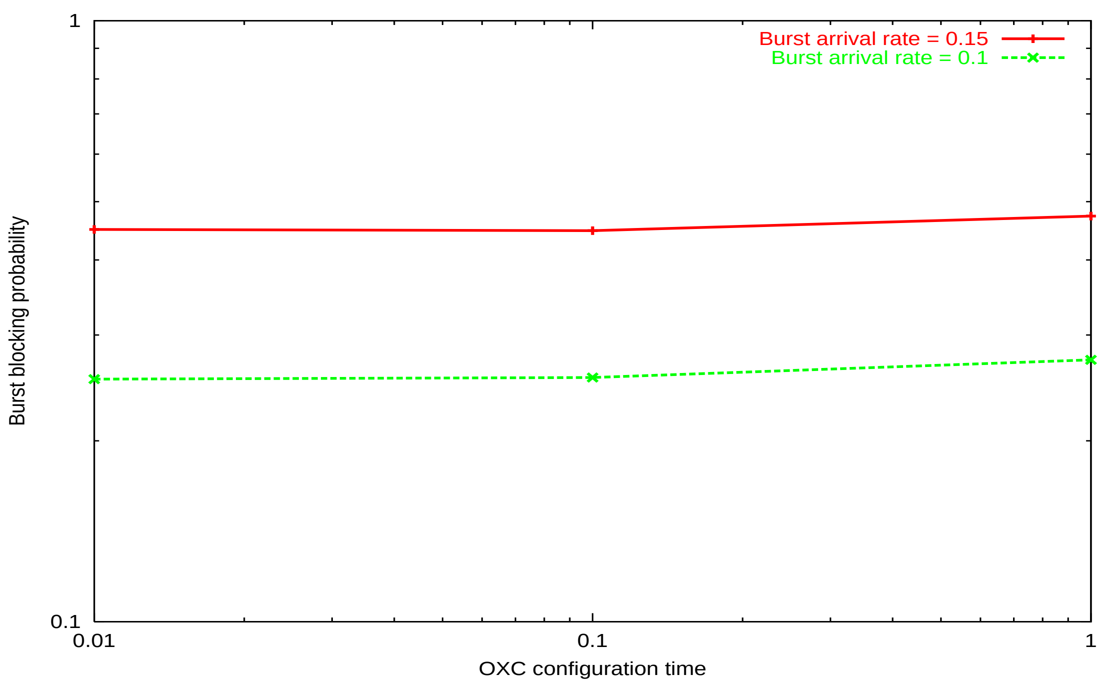

Figure 3.23: Burst drop probability of 5-hop paths, $C = 4$ converters

Chapter 4

Wavelength Assignment in OBS Networks

In this chapter, we study wavelength assignment problem in OBS networks. A fundamental assumption underlying most studies of optical burst switched (OBS) networks is that full wavelength conversion is available throughout the network. The existence of wavelength conversion capability at optical switches has a profound effect on the performance of an OBS network, since it removes the wavelength continuity constraint. Without wavelength conversion, a switch can forward an incoming burst to an output port *if and only if* the wavelength carrying the burst is available (free) on the output port. Otherwise, wavelength contention arises and the incoming burst is dropped. By allowing a switch to forward an incoming burst to an output port as long as the port has at least one free wavelength, full wavelength conversion eliminates the wavelength continuity constraint altogether, and improves significantly the performance of the OBS network. Currently, wavelength converters are expensive and complex devices, and this state of affairs is expected to continue in the foreseeable future. Therefore, it is widely expected that any wavelength conversion capabilities in the optical network will be limited and only sparsely deployed [34]. This observation has two important consequences. First, any performance studies relying on the assumption of full wavelength conversion will underestimate the burst drop probability in the network, possibly by a substantial factor, and may also fail to correctly identify the real behavior and dynamics of the network. Second, the absence of (full) conversion necessitates the development of good and efficient wavelength assignment policies. Such policies are even more important in OBS networks than in wavelength routed (circuit-switched) optical networks, due to the fact that in the former, a burst is transmitted without first reserving resources along the path. Therefore, a burst may be dropped at any intermediate switch along its path, even as it enters its last hop before the destination, resulting in substantial waste of network resources.

Although there is a substantial amount of research addressing the wavelength assignment problem in circuit-switched optical networks (for instance, refer to [61] and references thereof), the same problem has received little attention in the context of OBS networks. Recently, a priority-based wavelength assignment (PWA) algorithm was presented in [42]. Another recent work presented an algorithm to reduce wavelength contention in the OBS network by using some information regarding the routing paths [30]. Although these studies represent a step in the right direction, we feel that the issue of wavelength assignment in OBS network has not been adequately addressed, and that much remains to be done in order to develop a good understanding of the problem in all its aspects.

In this chapter, we attempt to fill this gap by presenting a comprehensive study of wavelength assignment in OBS networks. Specifically, we develop a suite of wavelength selection policies and we evaluate their relative performance in terms of both burst drop probability and fairness with respect to burst path lengths. Our policies attempt to alleviate the effects of wavelength contention by using the wavelength dimension to isolate traffic from different sources that uses overlapping paths through the network. We present two methods to achieve traffic isolation: the first is based on traffic engineering approaches that take into account the network topology and the routing paths to reduce wavelength contention though traffic isolation, while the second uses adaptive selection strategies that respond to feedback from the network. We also show that by appropriately combining the two methods we can achieve substantial improvement in performance.

Traffic engineering is proposed originally in TCP/IP, and used with multiprotocol label switching (MPLS), to make use of the network resource in an efficient way, to avoid unnecessary congestion in the network [37, 45]. Generalized MPLS (GMPLS) extends traffic engineering beyond packet switched networks [7], and it can be adopted to the domain of optical networks. So far, most of research of traffic engineering in optical networks is done in circuit switched WDM networks, and in routing and system management [26], traffic grooming [60], and restoration [13]. However, few works of traffic engineering have been done on OBS.

The next section discusses our main assumptions regarding the OBS network we study. In Section 4.2, we explain why conventional wavelength assignment schemes are not appropriate for OBS networks, and we develop a traffic engineering approach to achieve traffic isolation. In Section 4.3, we present a number of adaptive, priority-based wavelength assignment schemes, and we show how to combine them with the traffic engineering approach. We present the results of an experimental study of the performance of the various wavelength assignment policies in Section 4.4, and we conclude the chapter in Section 4.5.

4.1 The OBS Network Under Study

We consider an OBS network with N switches, interconnected in a general topology. Each link in the network can carry burst traffic on any wavelength from a fixed set of W wavelengths, $\{\lambda_1, \lambda_2, \cdots, \lambda_W\}$. The network switches employ the JIT reservation scheme [43] and the associated Jumpstart signaling protocol [5, 4] for JIT OBS networks. We emphasize, however, that the wavelength assignment policies we develop and evaluate in this work are independent of the specifics of the reservation protocol, and can be deployed alongside either the JET [33] or the Horizon [39] reservation schemes.

We assume that there are no wavelength converters in the OBS network; we emphasize, however, that our work can be extended to OBS networks with sparse conversion capabilities. A switch wishing to transmit a burst selects a free wavelength on the outgoing link for the transmission. The optical signal carrying the burst must then remain in the same wavelength on all the links along the path to the destination, unless an intermediate switch is capable of wavelength conversion. A wavelength contention arises when two bursts, which overlap in time, arrive at a switch on the same wavelength and need to use the same output port (outgoing link). We assume that switches have no buffers (electronic or optical) to store bursts; therefore, if the switch does not have any wavelength converters, one of the overlapping bursts is dropped. Consequently, wavelength selection at the source of the burst will critically affect the performance of the network in terms of burst drop probability.

The set of rules used by a switch in selecting the wavelength on which to transmit a burst define a wavelength assignment policy. Wavelength assignment is a hard problem that has been studied extensively in the context of wavelength routed networks [61]. Since wavelength assignment decisions must be made in real time, an efficient implementation approach is to have each switch order the W wavelengths in a wavelength list. When a switch has a new burst to send, it starts at the top of the list and transmits the burst on the first wavelength that is free on the desired outgoing link. Typically, all switches in the network will use the same policy (rules) to order the wavelengths. However, if the policy rules use information on the state of the network to rank wavelengths, the wavelength list at any given time may be different at various switches; furthermore, the wavelength list at a given node may change with time. This operation may result in different choices in wavelength assignment at various switches, and over time at the same switch. We also note that, a wavelength assignment policy is fully defined by describing the set of rules the network switches use to rank wavelengths.

We can classify wavelength assignment policies as adaptive or non-adaptive. In adaptive policies, the rules for ordering wavelengths take into account the network and traffic dynamics, hence, the order in which a given switch considers the wavelengths in search of a free one may change over time. In non-adaptive schemes, on the other hand, the order in which wavelengths are considered by each switch is neither dependent on, nor determined by, the prevailing network conditions. We emphasize that the rules of a non-adaptive policy may dictate a different wavelength list at different switches, or even a different ordering of wavelengths at a given switch over time; however, the rules must be independent of the network dynamics, although they may depend on certain properties of the network, such as topology or routing, that change at longer time scales.

Adaptive wavelength assignment policies depend on feedback from the network in order to adapt their rules to reflect the state of the network. This feedback can take many forms, depending on the specifics of the signaling protocol and the implementation details. The Jumpstart signaling protocol for JIT OBS networks [5, 4] provides such feedback in the form of two messages. The `setup ack` message is returned to the source of a burst by the destination switch, and indicates that the burst transmission was successful. The `reject` message is sent to the source of a burst by an intermediate switch when the latter is forced to drop the burst; certain fields of the `reject` message indicate the reason for dropping the burst, e.g., “destination unreachable” or “output port unavailable.” As a result, the source of a burst can determine whether the burst is successfully received or dropped, and in the latter case, where the drop occurred and whether the cause was wavelength contention. Some of the wavelength assignment policies we develop in this work rely on similar feedback from the network to adapt their rules.

In this chapter, we only consider the wavelength assignment problem. For simplicity, we assume fixed-path routing, in that all bursts between a source-destination pair follow the same path. Our work does not preclude changes in the routing paths, however, we make the reasonable assumption that any such changes take place at time scales significantly longer than the diameter of the network. Also, although we do not assume alternate routing, it is possible to modify the policies we develop to work when multiple paths are available between each source-destination pair.

4.2 Non-Adaptive Wavelength Assignment Schemes

4.2.1 First-Fit and Random

The First-Fit and Random wavelength assignment schemes are well-known and have been extensively studied in the context of wavelength routed networks [61]. We consider them here as baseline policies for comparing against the new schemes we develop.

In First-Fit, the W wavelengths are labeled arbitrarily and are listed in increasing order of label value, say, $\lambda_1, \lambda_2, \cdots, \lambda_W$. This order is identical at all network switches, and remains unchanged throughout the operation of the network. When a switch wishes to select a free wavelength for transmitting its burst, it searches the wavelength list in this order, until either a free wavelength is found and assigned to the burst, or the list is exhausted (in which case, we assume that the burst is dropped).

The Random wavelength assignment policy works as follows. We assume that each switch maintains a list of the wavelengths that are busy on each of its outgoing links. Suppose that at a given time, a switch needs to select a wavelength for a burst whose outgoing link has W' free wavelengths, $W' \leq W$. If $W' = 0$, the switch drops the burst; otherwise, it randomly allocates one of the W' free wavelengths to the burst. Note that this policy is within the class of policies we described in the previous section: each time a switch needs to make a selection, it lists the W wavelengths in some arbitrary order, and picks the first free wavelength in the list. However, this is a non-adaptive policy since the order is independent of the network state.

It is known that, in wavelength routed (i.e., circuit-switched) networks, where wavelength assignment decisions are based on complete knowledge of wavelength availability along the links of the path, First-Fit minimizes wavelength fragmentation and, hence, performs significantly better than Random in terms of blocking probability [61]. First-Fit is also simple to implement and does not require the exchange of any information among network switches regarding wavelength usage statistics.

However, in OBS networks, a switch must select a wavelength *without* any knowledge of the instantaneous wavelength occupancy of the links along the path. In this context, the First-Fit policy may in fact result in poor performance in terms of burst drop probability. In order to illustrate the problems associated with the First-Fit policy in OBS networks, consider the simple network shown in Figure 4.1. In this network, switches S_1 and S_2 transmit bursts which must travel over the common link e_3. The switches make wavelength assignment decisions using only local information, without any knowledge of the state of the link e_3 [1]. Since both switches search for a free wavelength in the same order, it is highly likely to pick the same wavelength, causing one of the bursts to be dropped at switch S_3. With the Random policy, on the other hand, the probability that both switches will select the same wavelength for the transmission is lower, leading to better performance. The performance results we present in Section 4.4 confirm this intuition; in fact, our study indicates that First-Fit is the worst policy by far, while Random performs significantly better in relative terms.

4.2.2 First-Fit-TE: Combining First-Fit and Traffic Engineering

We now present a modified version of the First-Fit wavelength assignment policy which is designed to overcome the shortcomings of the conventional First-Fit policy in OBS networks. In order to motivate our approach, let us return to the scenario depicted in Figure 4.1, and assume again that the W wavelengths on each link are labeled $\lambda_1, \cdots, \lambda_W$. It is not difficult to see that, among all wavelength assignment policies that use only local information at switches S_1 and S_2, the following policy would minimize the burst drop

[1]Note that, due to the relatively short duration of bursts, any information that switches S_1 and S_2 may have regarding the state of link e_3 may already be out-of-date by the time they receive it; therefore, such information will not be useful in making wavelength assignment decisions.

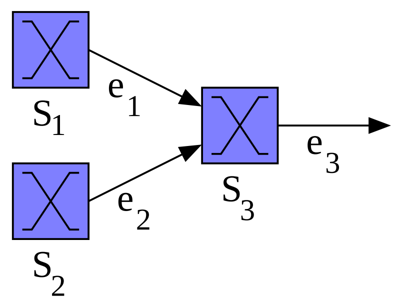

Figure 4.1: First-Fit results in high burst drop probability at Switch S_3

probability at switch S_3:

- One of the two switches (say, S_1) uses the First-Fit policy, and searches for a free wavelength in the order $\lambda_1, \lambda_2, \cdots, \lambda_W$.
- The other switch (say, S_2) also uses the First-Fit policy, but searches for a free wavelength in the reverse order $\lambda_W, \lambda_{W-1}, \cdots, \lambda_1$.

This policy minimizes the burst drop probability at switch S_3 because switches S_1 and S_2 will select the same wavelength (and thus, a burst will be dropped at switch S_3) if and only if all other wavelengths are busy transmitting bursts. In contrast, other policies using only local information at switches S_1 and S_2 (e.g., Random, conventional First-Fit, etc.) might select the same wavelength at both switches even while other wavelengths are free.

While it is straightforward to identify the optimal wavelength assignment policy for the simple network of Figure 4.1, determining the optimal policy for a large network with a general topology is a difficult and complicated task. Therefore, we now present a new wavelength assignment policy that is similar to First-Fit, but uses information regarding the network topology and routing paths to improve upon conventional First-Fit in terms of the burst-drop probability; we will refer to this new policy as First-Fit-TE, where "TE" stands for "traffic engineering."

Consider an OBS network with general topology. The network consists of N switches, and each link can carry W wavelengths. The W wavelengths are labeled arbitrarily as $\lambda_1, \cdots, \lambda_W$, and this order is fixed and known at all N switches. Each switch $S_i, i = 1, \cdots, N$, is assigned a *start wavelength*, $start(i) \in \{\lambda_1, \cdots, \lambda_W\}$. The value of $start(i)$ is determined using a traffic engineering approach we describe shortly, and remains fixed throughout the operation of the network[2]. Furthermore, it is possible that two different switches, S_i and $S_j, j \neq i$, be assigned the same start wavelength, $start(i) = start(j)$.

The First-Fit-TE wavelength assignment policy at switch $S_i, i = 1, \cdots, N$, operates as follows:

- When the switch has a new burst to transmit, it searches for a free wavelength in the order:

$$\lambda_{start(i)}, \lambda_{start(i)+1}, \cdots, \lambda_W, \lambda_1, \cdots, \lambda_{start(i)-1}.$$

- The switch transmits the burst on the first free wavelength found, and drops it if all W are found busy.

In other words, each switch follows a First-Fit policy, but, unlike the conventional First-Fit scheme that requires all nodes to use the same search sequence, under First-Fit-TE, the starting wavelength of the search sequence can be different for different switches.

Let $d(i, j)$ denote the distance between the start wavelengths of the two switches S_i and S_j in the sequence $\lambda_1, \cdots, \lambda_W$:

$$d(i, j) \quad = \quad start(j) \ominus start(i)$$

where $\ominus$ denotes subtraction modulo-W. We note that, when the network is not heavily loaded, the wavelengths on which a switch S_i transmits its bursts will be close to its start wavelength $start(i)$. Therefore, the main idea behind the First-Fit-TE policy is to assign a start wavelength to each switch in the network in such a manner that, the higher the "interference" among bursts originating at two switches S_i and S_j, the higher the distance $d(i, j)$ between the start wavelengths of the two switches. In this context, we use the notion of "interference" as a measure of the likelihood that bursts generated by different switches will use the same link on the way to their respective destinations.

The level of "interference" among two switches depends on the network topology, the relative location of the switches in the network, the traffic characteristics and the routing algorithm. For instance, bursts from two switches located at diametrically opposite points in a large network are likely to use non-overlapping paths, while bursts originating at two neighboring switches may use paths with substantial overlap; we say that the former pair

[2]Note that it is possible to update periodically the values of $start(i), i = 1, \cdots, N$, to reflect changes in the network topology and/or routing paths. However, we expect that any such updates will take place over long time scales and will have only a transient effect on the network operation.

of switches has low "interference" while the latter pair has high "interference." We now formalize the concept of interference in a quantitative manner. In the following discussion, we assume that the network employs fixed routing so that bursts between a given source-destination pair always follow the same path; however, our main idea can be adapted to apply to other routing schemes.

Let Π_i denote the set of paths taken by bursts originating at switch S_i, $\Pi_i = \{\pi_{i1}, \pi_{i2}, \cdots \pi_{iN}\}$, where π_{ij} is the path from switch S_i to switch S_j. Let also γ_{ij} denote the traffic load of bursts from switch S_i to switch S_j. We define the *degree of interference* of a path π_{ij} and a switch S_k, denoted by $ID(\pi_{ij}, k)$, as the amount of traffic from switch S_i to S_j on the path π_{ij} that interferes with traffic originating from switch S_k:

$$ID(\pi_{ij}, k) = \begin{cases} \gamma_{ij}, & \pi_{ij} \text{ shares a link with a path in } \Pi_k \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

We also define the *interference level* between two switches S_i and S_j, which we will denote by $IL(i, j)$, as:

$$IL(i, j) = \begin{cases} \sum_{\pi_{ik} \in \Pi_i} ID(\pi_{ik}, j) & i \neq j \\ 0, & i = j \end{cases} \tag{4.2}$$

That is, $IL(i, j)$ is the total amount of traffic originating at switch S_i which may interfere (through the use of common network links) with any traffic originating at switch S_j. Finally, we define the *combined interference level* $CIL(i, j)$ between two switches S_i and S_j as the total interference between the two switches:

$$CIL(i, j) = IL(i, j) + IL(j, i), \quad i \neq j \tag{4.3}$$

With the above definitions, the higher the combined interference level between two switches, the higher the likelihood that bursts from the two switches will share some network link. Therefore, to minimize the probability that bursts from the two switches will collide on a common link, we must ensure that they do not use the same wavelength. In other words, we must assign start wavelengths to the two switches that are far apart from each other. Conversely, if the interference level between two switches is low, their bursts are less likely to share links and collide; consequently, the start wavelengths of the two switches can be close to each other.

Given the interference levels $IL(i, j)$ for all pairs of switches (S_i, S_j) in the network, our objective is to determine the start wavelength $start(i)$ for each switch S_i so as to minimize the burst dropping probability in the network under the First-FIT-TE wavelength assignment policy defined earlier. It might be tempting to formulate this problem as an integer optimization problem and attempt to solve it using standard problem solvers. Unfortunately, it is not possible to express the objective function (i.e., the network-wide burst drop probability) analytically in terms of the problem variables. Even if we chose to formulate the problem in terms of a different objective function for which such an analytical expression is available, two issues would arise. First, there is the question of what would be an appropriate and relevant objective function; and second, even if we were to find an appropriate objective function, the complexity of the resulting problem would preclude the use of optimal solution methods for anything other than small, toy networks.

Instead, we use a simple heuristic to assign start wavelengths to the various switches, which we have found to work well in practice. The heuristic consists of three steps:

1. Partition the set of N switches in K groups (subsets), $g_1, g_2, \cdots, g_K$, such that there is little interference among switches in each group. All the switches in a given group $g_k, k = 1, \cdots, K$, will be assigned the same start wavelength.

2. Arbitrarily label the W wavelengths as $\lambda_1, \cdots, \lambda_W$, and let $x = W/K$ (note that x may not be integer). We evenly space the K start wavelengths across the W wavelengths, such that the k-th start wavelength is the wavelength labeled $\lambda_{1+\lfloor (k-1)x \rfloor}$.

3. We assign the K start wavelengths to each of the K groups so as to minimize the interference level among groups with adjacent start wavelengths.

Let us now explain the first and third steps of the heuristic in more detail.

Partitioning. Typically, partitioning problems with objective functions similar to the one we consider here (i.e., to minimize the interference among switches in each group) are hard optimization problems [21]. Therefore, we use the following greedy heuristic to assign each switch to one of K groups. Let $N = LK + M, M < K$; in our heuristic, the first M groups will consist of $L + 1$ switches, and the last $K - M$ groups of L switches. Consider group $g_k, k = 1, \cdots, K$. Initially, $g_k = \emptyset$. Select the switch S_i that has not been assigned to a group yet, such that S_i has the minimum total combined interference level, $\sum_{j=1}^{N} CIL(i, j)$, among unassigned switches. Let $g_k \leftarrow g_k \cup \{S_i\}$. Then, select the unassigned switch S_j that has the minimum combined interference level $CIL(i, j)$ with switch S_i, and let $g_k \leftarrow g_k \cup \{S_j\}$.

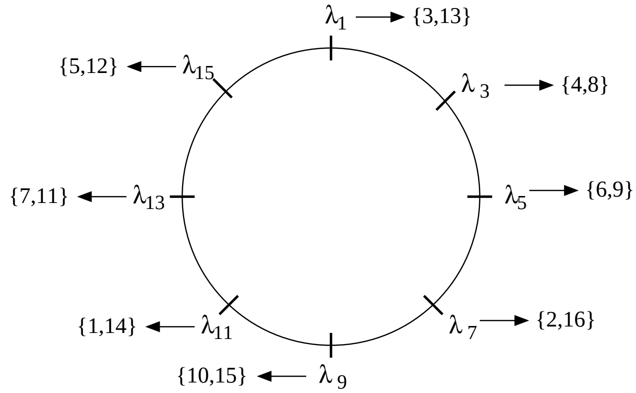

Figure 4.2: Start wavelength for each group of switches, NSFNet, $W = 16$ wavelengths, $K = 8$ groups

Continue in this manner, selecting the next switch to add to g_k so as to minimize the overall combined interference level in the group, until the total number of switches in group g_k has been reached. If $k = K$, the algorithm stops; otherwise, it continues with group g_{k+1}.

Assignment of start wavelengths. Again, we use a greedy algorithm to assign start wavelengths to groups of switches in sequential order. First, note that the first start wavelength is always λ_1. We assign this wavelength as the start wavelength of the group g for which the total combined interference level among all switches in g and switches in any other group is minimum (over all K groups). Suppose now that the first $k, k < K$, start wavelengths have been assigned, and let g be the group that was the last to be assigned a start wavelength. Let g' denote the unassigned group such that the total interference among switches in g and switches in g', $\sum_{S_i \in g, S_g \in g'} IL(i,j)$, is minimum. Then, we assign the $(k+1)$-th start wavelength to group g'. The algorithm proceeds in this manner until all groups have been assigned start wavelengths.

To illustrate our approach, let us consider the two network topologies shown in Figures 4.3 and 4.4. To simplify the presentation, we assume that the traffic load $\gamma_{ij} = \gamma = 1$ for all switch pairs (S_i, S_j). The 4×4 torus network of Figure 4.3 has a regular topology, and is a dense network with each node having a rather high degree. The 16-node network of Figure 4.4, on the other hand, has an irregular topology which is obtained by augmenting the 14-node NSFNet topology through the addition of two fictitious switches, switch S_1 and switch S_{16}, to capture the effect of NSFNet's connections to Canada's communication network, CA*net.

For each network topology, we first run Dijkstra's algorithm to compute the shortest path for each pair of switches. We then computed the interference level $IL(i,j)$ for each pair of switches (S_i, S_j) using expressions (4.2) and (4.1), after letting $\gamma_{ij} = 1$ for all i, j. Tables 4.1 and 4.2 list the interference levels for each pair of switches in the torus and NSFNet topologies respectively. Assuming that $K = 8$, i.e., that we partition the 16 switches into 8 groups of size 2, the groups for the torus network are: $\{1, 11\}$,{2,12}, $\{3, 9\}$,{4,10}, $\{5, 15\}$,{6,16}, $\{7, 13\}$, and $\{8, 14\}$. For the NSFNet, on the other hand, the eight groups are: $\{1, 14\}$, $\{2, 16\}$, $\{3, 13\}$, $\{4, 8\}$, $\{5, 12\}$, $\{6, 9\}$, $\{7, 11\}$, and $\{10, 15\}$. Also, the start wavelength assigned to each group of switches in the NSFNet is shown in Figure 4.2.

Finally, we note that the approach we presented in this section assumes that the paths π_{ij} between all pairs of switches in the OBS network are given, and computes the interference levels as in expressions (4.1)-(4.3). An interesting problem, which is outside the scope of this thesis, is to compute the paths so as to minimize the interference levels among the various switches. We are currently working on routing algorithms for OBS networks that take interference into account when computing paths.

4.3 Adaptive Wavelength Assignment Schemes

In adaptive wavelength assignment schemes, the order in which each switch uses to search for an available wavelength changes over time in response to the state of the network and prevailing traffic conditions. A common mechanism to implement adaptive wavelength assignment, which we adopt in this work, is by assigning a *priority* to each wavelength. At any given instant, the priority of a wavelength reflects the likelihood that a burst transmission on this wavelength will be successful, i.e., the burst will not be dropped due to wavelength contention at an intermediate switch. The wavelength priorities are updated periodically based on feedback from the network, so as to reflect the current network conditions. Specifically, when a switch determines that a burst transmitted on a particular wavelength has been successfully received, it increases the priority of the wavelength; conversely, if the burst is dropped inside the network, the priority of the wavelength carrying the burst is decreased. Typically, every switch in the OBS network uses the same algorithm to set the priority of wavelengths, and maintains locally a list of the W wavelengths in de-

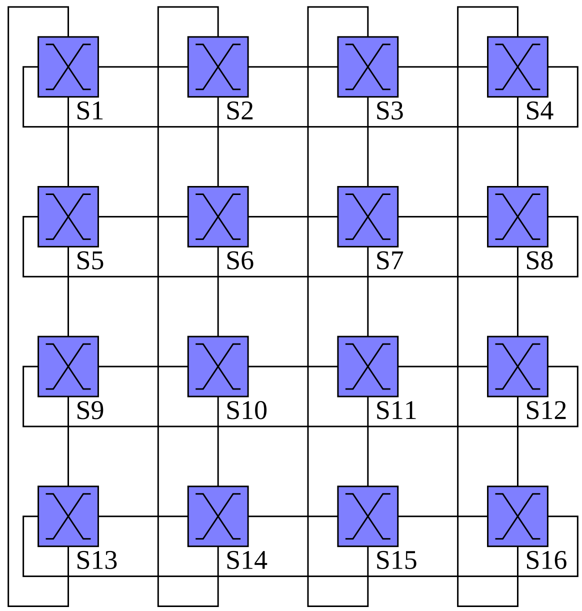

Figure 4.3: The 4×4 torus network

Table 4.1: Interference levels $IL(i, j)$ for the 4×4 torus network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	12	6	11	11	9	4	6	4	2	0	3	14	5	4	4
2	12	0	10	8	9	11	8	6	4	10	5	0	6	14	12	1
3	5	8	0	14	3	6	10	9	0	7	10	6	3	8	10	11
4	14	11	12	0	8	7	9	12	3	0	5	10	7	6	9	12
5	9	8	4	6	0	12	6	9	11	3	1	9	8	2	0	1
6	5	14	6	6	14	0	8	8	7	10	3	4	3	7	6	0
7	2	7	14	12	7	14	0	14	2	4	10	10	0	3	5	8
8	8	6	12	14	14	12	14	0	11	2	10	14	6	0	4	12
9	8	2	0	2	9	4	1	5	0	8	3	14	10	4	3	4
10	7	12	5	0	9	7	5	3	14	0	9	7	11	14	10	1
11	0	2	10	5	3	7	14	12	3	14	0	11	0	12	12	9
12	3	0	7	8	5	3	10	11	11	7	12	0	4	2	7	14
13	12	6	4	6	3	1	0	1	8	3	0	7	0	5	5	12
14	11	13	10	7	6	3	3	0	9	13	6	2	11	0	13	1
15	4	6	12	8	0	2	6	3	2	12	10	6	4	13	0	5
16	7	2	8	11	1	0	8	8	3	1	9	14	6	2	7	0

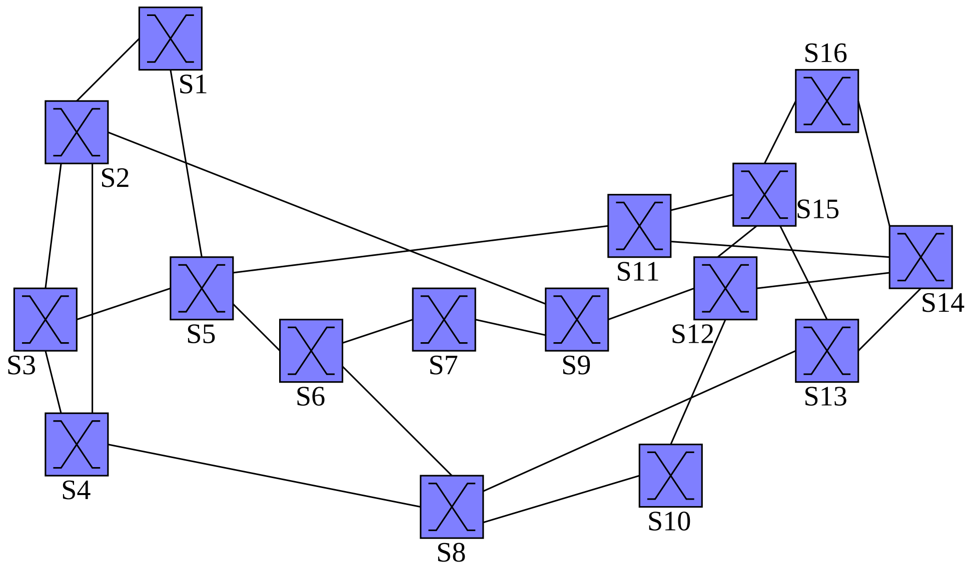

Figure 4.4: The 16-node topology based on the 14-node NSFNet

Table 4.2: Interference levels $IL(i,j)$ for the 16-node NSF network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	13	13	9	3	3	11	14	13	5	3	4	3	1	12	3
2	13	0	13	14	7	7	6	3	2	2	9	3	2	3	11	2
3	13	13	0	6	11	14	5	3	3	13	1	2	1	13	11	1
4	11	12	8	0	14	10	10	4	2	3	14	11	10	6	14	11
5	5	9	8	14	0	14	14	13	1	11	5	5	4	13	13	4
6	5	9	14	7	14	0	8	4	1	14	4	5	4	14	7	4
7	9	7	3	5	14	13	0	14	13	7	3	6	5	4	9	5
8	14	8	6	3	3	2	14	0	14	9	3	11	10	3	9	11
9	13	2	4	2	1	1	13	14	0	14	12	12	11	10	4	12
10	6	2	13	2	13	14	7	7	14	0	4	8	6	10	2	7
11	4	11	1	14	6	4	4	4	12	4	0	12	11	8	13	12
12	5	3	3	10	5	4	5	10	11	11	13	0	10	14	5	14
13	3	1	1	6	3	2	3	6	11	5	12	10	0	8	3	14
14	1	3	13	5	13	14	3	4	3	10	12	13	10	0	2	12
15	14	11	8	14	8	6	9	6	4	2	13	6	5	2	0	5
16	3	1	1	8	3	2	3	9	11	8	12	14	14	11	3	0

creasing order of priority. Therefore, the order in which a given switch considers wavelengths for burst transmission changes over time, according to the relative changes in wavelength priorities. Furthermore, at any given time instant, the wavelength order at one switch may be different, possibly substantially so, than the wavelength order at another switch, due to the differences in transmission success that bursts from the two switches experience over the various wavelengths.

A wavelength assignment scheme based on priorities was presented in [42], and was referred to as "priority wavelength assignment" (PWA). This work assumes a single, fixed path for each source-destination pair (S_i, S_j) which all bursts from switch S_i to S_j follow. Under PWA, each switch S_i in the OBS network maintains locally a priority value for each wavelength-destination pair; in other words, switch S_i assigns a priority to each tuple $(\lambda_w, S_j), w = 1, \cdots, W, i \neq j = 1, \cdots, N$. The priority of tuple (λ_w, S_j) is set to the ratio of the number of bursts which have been successfully transmitted from S_i to S_j on wavelength λ_w (along the fixed path associated with this pair of switches) over the total number of bursts transmitted from S_i to S_j on the same wavelength. When switch S_i needs to transmit a burst to S_j, it considers the wavelengths in decreasing order of priority of the corresponding tuples (λ_w, S_j) and uses the first free one. Depending on the outcome of the transmission, the switch then updates the priority of the tuple. It is shown in [42] that, under low load, PWA performs better than the Random wavelength assignment policy in terms of burst drop probability; under high load, on the other hand, it performs only marginally better than Random.

We now introduce two additional PWA schemes which differ from the the one presented in [42] in two ways. First, a priority is associated with each wavelength in a different way than in [42], resulting in a trade-off between complexity (in both space and time) and performance. Second, our notion of priority, and the manner it is incremented and decremented, are different than the one in [42]. Next, we describe the operation of the new PWA schemes, and then we define the priority values and the way they are updated. In our discussion, we will use the notation $p(\bullet)$ to denote the priority function.

The first scheme, which we call "PWA-Link," works as follows. Each switch S_i maintains a priority value for each wavelength-link pair, i.e., for each tuple $(\lambda_w, e), w = 1, \cdots, W, e \in E$, where E is the set of links in the network. Whenever the switch wishes to transmit a burst to some switch S_j over path $\pi = \{e_1, e_2, \cdots, e_k)$, it computes the wavelength-path priorities $p(\lambda_w, \pi)$ by adding up the corresponding wavelength-link priori-

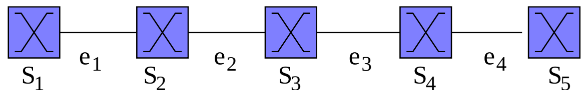

Figure 4.5: A linear network to illustrate the difference between PWA and PWA-Link

ties along the path links:

$$p(\lambda_w, \pi) \quad = \quad \sum_{e \in \pi} p(\lambda_w, e) \qquad w = 1, \cdots, W$$

The switch considers the wavelengths in decreasing order of $p(\lambda_w, \pi)$, and transmits the burst on the first free wavelength [3]. Upon learning the outcome of the transmission, the switch:

- *increments* the priority of the links, if any, on which the burst was successfully transmitted;
- *decrements* the priority of the link, if any, at which it was dropped due to contention; and
- *maintains* the priority of any other links (e.g., links following the one where the burst was dropped).

We will explain shortly how the priorities are incremented or decremented.

PWA-Link operates at finer granularity and uses more information than PWA in making wavelength assignments, therefore one might expect that it would lead to better performance; indeed, numerical results to be presented in the next section confirm that a network employing PWA-Link has lower overall burst drop probability than when pure PWA is employed. To explain the difference in performance, let us consider the simple linear network shown in Figure 4.5, and suppose that switch S_1 transmits a burst to switch S_5 on some wavelength λ_w. Suppose further that the burst is dropped at switch S_4. Under PWA, the priority of the tuple (λ_w, S_5) is decremented, without taking into account the fact that the burst transmission was successful on the first three links of the path from S_1 to S_5; indeed, $p(\lambda_w, S_5)$ is decremented by the same amount regardless of which switch in the path dropped the burst. In PWA-Link, on the other hand, this additional information is used

[3] Throughout this chapter, we assume that ties are broken arbitrarily, so that a switch may select any one of a set of wavelengths having the same priority with uniform probability.

in updating the priorities of the wavelength-link tuples. Since the burst was successful on links e_1, e_2, and e_3, the priorities $p(\lambda_w, e_1)$, $p(\lambda_w, e_2)$, and $p(\lambda_w, e_3)$ are incremented, while $p(\lambda_w, e_4)$ is decremented. Note that by increasing the priorities of λ_w on the first three links, this wavelength will move up the list with respect to burst transmissions to switches S_2, S_3, and S_4, as it should, since the burst reached all three switches successfully.

The second scheme we propose is simpler than both PWA and PWA-Link, and we will refer to it as "PWA-λ." With this scheme, each switch S_i assigns a priority value $p(\lambda_w)$ to every wavelength $\lambda_w, w = 1, \cdots, W$. When switch S_i successfully transmits a burst on wavelength λ_w, the priority $p(\lambda_w)$ is incremented regardless of the destination of the burst or the path traveled. Otherwise, the priority of the wavelength is decremented. Intuitively, PWA-λ will perform worse than either PWA or PWA-Link in terms of burst drop probability, but it is simpler and easier to implement.

Let us now consider the space and time complexity of implementing the three PWA schemes at each switch. PWA-λ requires $O(W)$ memory to record the priority information, where W is the number of wavelengths. It also needs $O(1)$ time to update the priority value of a wavelength once the relevant feedback from the network has been received, and $O(\log W)$ time to maintain a sorted priority list. PWA requires $O(WN)$ memory for recording priority values, where N is the number of switches in the network; and it takes constant time to update the priority of a wavelength-destination pair. It also takes $O(\log W)$ time to maintain a sorted priority list, since it only needs to have one such list of W elements (wavelengths) for each of the N destinations. Finally, PWA-Link needs $O(W \mid E \mid)$ memory for the priority values, where E is the set of links in the network. When the feedback regarding a burst transmission is received, the switch must update the priority of all wavelength-link pairs along the path, and this operation takes time $O(\Delta)$, where Δ denotes the diameter of the network. The computational overhead for maintaining sorted lists for all the destinations is $O(N' \log W)$, where N' is the number of paths containing the links whose priorities with a wavelength are just updated. As we can see, the three PWA schemes represent a tradeoff between implementation complexity and performance, with PWA-Link being the best performing but most complex, PWA-λ the worst performing but easiest to implement, and pure PWA occupying the middle ground in both metrics.

We now turn our attention to the priority function and the increment and decrement operations used to update the wavelength priorities. Recall that in [42], which introduced PWA, the priority of a wavelength-destination (λ_w, S_j) pair was defined as the fraction of transmissions to destination S_j on wavelength λ_w that have been successful. However, our experimental investigations indicate that this measure may not be appropriate because of disparities in the rates of change in priority over time and across wavelengths. Specifically, while initially the rate of change is relatively large, the rate of change diminishes over time: once the number of burst transmitted on a wavelength becomes relatively large, each additional transmission has a negligible effect on the priority, regardless of the outcome. As a result, once the network has been in operation for a while and the priorities have settled, it will take a long time for priorities to adapt to any changes in the traffic or network dynamics, during which bursts will use suboptimal wavelengths and the burst drop probability will be high. Furthermore, the rate of change in priority can be different for different wavelengths, possibly substantially so. For instance, consider two wavelengths that have the same priority but one has been used substantially more often that the other for transmitting bursts. In this case, at each step (i.e., burst transmission), the priority of the wavelength that has been used more frequently will change by a small amount in either direction, while the the priority of the less frequently used wavelength will change by a larger amount. Given that both these properties are undesirable, it would be preferable to use a priority scheme in which the rate of change at each update is not affected by length of time or frequency of use of a wavelength.

In our work, the priorities $p(\bullet)$ are taken to be real numbers in the range $[1, W]$, and are initialized to $W/2$. We use an "Additive Increase, Additive Decrease" (AIAD) scheme to update the priorities, with increment Inc and decrement Dec. Specifically, after the result of a burst transmission has become known, a switch takes the following steps.

- If the burst transmission was successful, the appropriate priority (or priorities, in the case of PWA-Link) are incremented as:
  $p(\bullet) \leftarrow \max\{p(\bullet) + Inc,\ W\}$.
- Otherwise, the appropriate priority(-ies) are decremented as:
  $p(\bullet) \leftarrow \min\{p(\bullet) - Dec,\ 1\}$.

We have conducted a large number of experiments to determine the best values of the increment Inc and decrement Dec to use with the AIAD scheme. Our results indicate that the performance of the PWA policies is best when $Inc < Dec$, and Inc takes values from $0.2 \sim 0.4$, while the value of Dec is in the range $0.8 \sim 1.2$.

4.3.1 Combining PWA and Traffic Engineering

The PWA schemes achieve low burst drop probability by using the wavelength dimension to *isolate* interfering bursts. To see this, consider the scenario when burst traffic originating at two different switches uses overlapping paths through the network. The two switches will initially experience high burst loss and will try several different wavelengths for their traffic. Note that, the dropping of a burst from one switch due to the presence of a burst from the other leads to a decrease of the wavelength's priority in the former switch and an increase of the same wavelength's priority in the latter switch. Eventually, the priority of wavelengths on which one switch has been successful in transmitting bursts will rise high enough that they become the preferred wavelengths for this traffic, while the same set of wavelengths will fall out of favor at the other switch (due to low priority); and vice versa. In effect, the traffic from one switch is isolated from the traffic of the other through the use of different sets of wavelengths.

Recall now that the First-Fit-TE scheme we presented in Section 4.2.2 also attempts to achieve traffic isolation. The difference is that it takes a traffic engineering approach, using information about the network topology, traffic demands, and routing paths to assign different start wavelengths to each switch. Therefore, it is natural to investigate whether combining the traffic engineering approach with the adaptive PWA schemes might produce further improvements in performance.

We now present a small modification to the PWA schemes to take advantage of the traffic engineering approach of Section 4.2.2. The modification is applied at initialization time only, while the operation of the PWA schemes remains identical to the one described above. Recall that, in the original PWA schemes, all priorities are initialized to $W/2$. Therefore, initially all wavelengths are indistinguishable from each other with respect to transmission preference. The modification we propose is to use different initial priorities at each switch, so that different switches will be forced to use different wavelengths for interfering traffic from the very beginning. If the initial values are determined appropriately, this approach has the following benefits over pure PWA: (1) the initial burst losses will be avoided; (2) the switches will settle to preferred wavelengths faster; and (3) the network will achieve better overall traffic isolation.

Similar to the First-Fit-TE wavelength assignment policy, we arbitrarily order the W wavelengths as $\lambda_1, \cdots, \lambda_W$, and we assign start wavelengths to the switches as we described in Section 4.2.2. Consider some switch S_i, and let $start(i)$ be its start wavelength. Let also $next$ denote the next wavelength (modulo-W) that is assigned as the start wavelength of another switch; in other words, the wavelengths $\lambda_{start(i)\oplus 1}, \cdots, \lambda_{next\ominus 1}$ are not assigned as start wavelengths for any switch ($\oplus$ and $\ominus$ denote addition and subtraction, respectively, modulo-W). Then, at switch S_i all priorities involving wavelengths $\lambda_{start(i)}, \cdots, \lambda_{next\ominus 1}$ are initialized to $W/2+Inc$, while the priorities of all other wavelengths are initialized to $W/2$, as before [4]. As a result, the switch will initially give preference to wavelengths $\lambda_{start(i)}, \cdots, \lambda_{next\ominus 1}$ when transmitting bursts. The operation of the PWA schemes is not affected in any other way.

We will use the terms "PWA-TE," "PWA-Path-TE," and "PWA-λ-TE to refer to the versions of PWA, PWA-Path, and PWA-λ in which wavelength priorities are initialized in the manner described above.

4.4 Numerical Results

In this section, we use simulation to compare the various static and adaptive wavelength schemes. We consider two 16-node network topologies, the 4×4 torus network shown in Figure 4.3, and the NSF network in Figure 4.4. Our goal is to compare the various wavelength assignment schemes in terms of (1) overall (network-wide) burst drop probability, and (2) burst drop probability as a function of path length. Since, in an OBS network without wavelength converters, the drop probability may increase with the number of hops a burst has to traverse, it is important that the wavelength assignment scheme achieve some degree of fairness among bursts that travel over paths of different length. In our simulations, the burst arrival process of each switch is Poisson and the burst length is exponentially distributed with mean $1/\mu$. For simplicity, we also assume that bursts originating at a given switch are equally likely to be destined to any of the other switches. We used the method of batch means to estimate the burst drop probability; each of the simulation runs lasts until 400,000 bursts have been transmitted by the whole network. We have also obtained 95% confidence intervals for all our results; however, they are so narrow that we omit them from the figures we present in this section in order to improve readability.

Figures 4.6 and 4.7 plot the burst drop probability of the wavelength assignment

[4] Note that adding Inc to the initial value of the priority of a wavelength is equivalent to assuming that a single burst has been successfully transmitted on that wavelength.

schemes we described in Sections 4.2 and 4.3 for the 4 torus network and for $W = 64$ wavelengths. Figure 4.6 shows the drop probability for low traffic load, while Figure 4.7 shows the performance of the network under moderate and high loads. From the figures, we see that First-Fit is always worse than Random, confirming our previous analysis. We also make two important observations. First, the adaptive, priority-based schemes (i.e., PWA, PWA-Link, and PWA-λ) perform better than Random (with the exception of PWA-λ at low loads less than 0.1). In particular, PWA-Link, which uses more detailed information than the original PWA, is the best of the three adaptive schemes, PWA is the second best, while PWA-λ, which uses the least amount of information, is the worst of the three. The second observation is that the traffic engineering approach we described earlier to achieve traffic isolation and reduce traffic interference, when combined with any wavelength assignment scheme, static or adaptive, leads to a significant decrease in burst drop probability. The most dramatic impact is with the First-Fit scheme, in which case First-Fit-TE has a burst drop probability that is up to two orders of magnitude lower than the plain First-Fit policy. Similar decreases (although of smaller magnitude) can be observed for the PWA-TE, PWA-Link-TE, and PWA-λ-TE schemes over the respective non-TE versions. Overall, we find that the best approach to wavelength assignment in OBS networks is to combine adaptive, priority-based schemes with our traffic engineering approach. Interestingly, we find that PWA-TE is the best performing scheme, having lower burst drop probability than even the PWA-Link-TE scheme over a wide range of load values (note that, in contrast, PWA-Link performs much better than PWA). We believe that this result is due to the fact that, in PWA-TE, the wavelength priorities are adjusted by considering the whole path of a burst, not individual links as in PWA-Link-TE, and this operation is more compatible with the traffic engineering approach we take.

Figures 4.8 and 4.9 are similar to the previous two figures, but compare the burst drop probability of the nine wavelength assignment schemes for the NSFNet topology. We note that the burst drop probability is higher than in the torus network for the given load. This result is due to the fact that the NSFNet topology is (1) more sparsely connected than the torus network, and (2) irregular, and thus, without the inherent load balancing properties of the torus topology. As a result, certain links may become heavily congested when using shortest path routing, leading to higher burst drop probability [5]. The relative

[5]As we mentioned earlier, in the case of sparsely connected, irregular topologies, computing paths so as to minimize traffic interference may lead to significant improvements over shortest path routing. Such

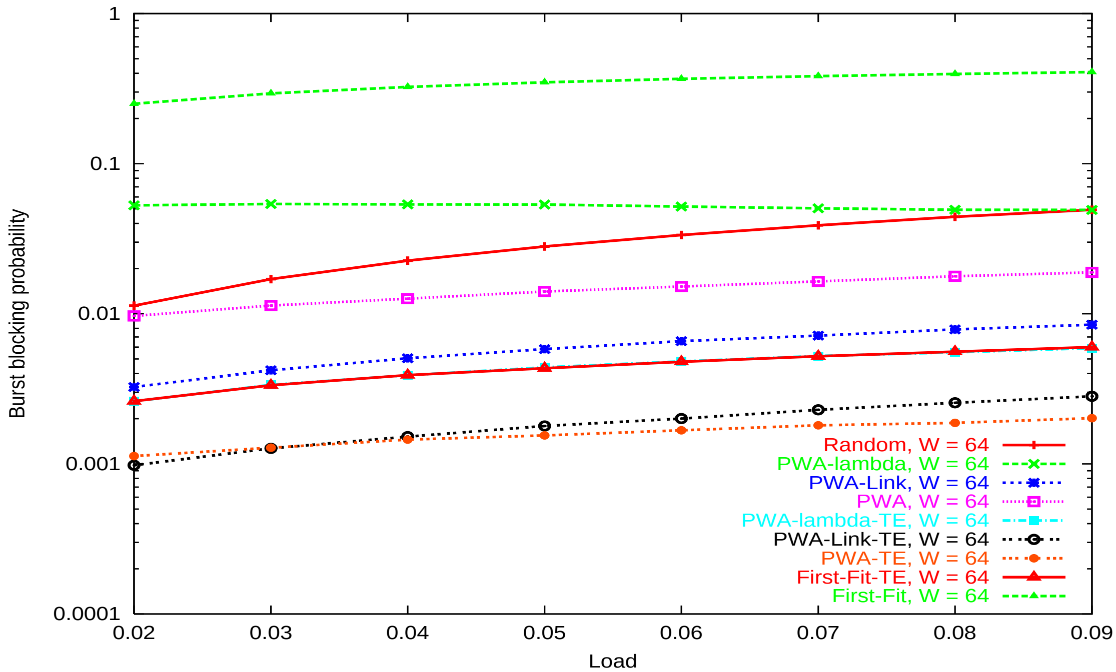

Figure 4.6: Burst drop probability, 4×4 torus network, low load

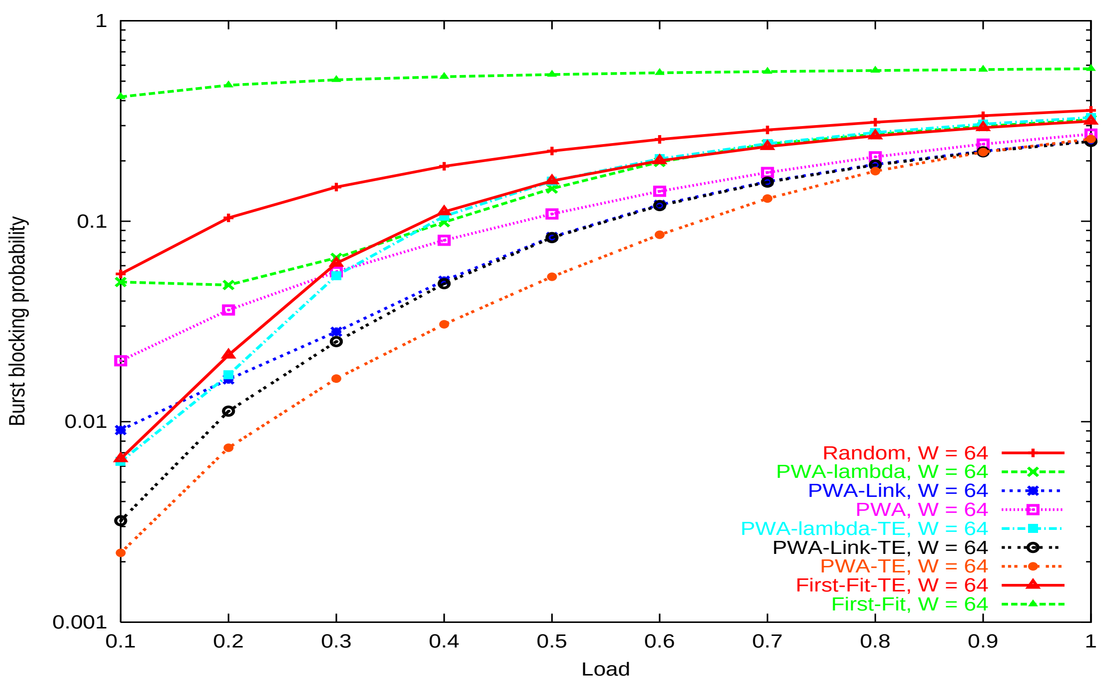

Figure 4.7: Burst drop probability 4×4 torus network, moderate and high load

performance of the wavelength assignment schemes is very similar to the one we observed for the torus network: adaptive, priority-based schemes are better than static ones, and incorporating traffic isolation through traffic engineering leads to a decrease in drop probability. As before, PWA-TE is the best policy overall, except at very low loads. We also note that, at very high loads, the performance of all policies is similar; this is due to the fact that, at such high loads, burst dropping is mostly due to the lack of wavelengths.

Now let us consider the gain as we increase the number W of available wavelengths. Figures 4.10 and 4.11 plot the burst drop probability of the nine schemes as the number of wavelengths increases from 8 to 128, for the torus and NSFNet, respectively. The load per wavelength in the network is kept constant at 0.2 for these experiments. As we can see, the burst drop probability of First-Fit increases, and it remains mostly unchanged in the case of Random. These results are expected: Random distributes the bursts randomly to the various wavelengths, but since the load per wavelength is constant, there is little change in overall drop probability; while First-Fit attempts to use the same few first wavelengths, thus an increase in overall load, as W increases, results in higher drop probability. For the other schemes, in general, the drop probability decreases with the number of wavelengths, up to a point. But the decrease is not as dramatic as other studies, which assume full wavelength conversion, have shown. This result indicates that OBS networks will benefit from some degree of wavelength conversion. A "strange" observation is that in Figure 4.11, First-Fit-TE becomes worse with the increase of W. The reason is: the increase of wavelengths widens the distance between the start wavelengths among different switches; however, for each switch, e.g., switch S_4, the increase of the overall load makes the utilization of the wavelengths near its start wavelength higher, i.e., λ_3,λ_4 in Figure 4.2; in this case, if switch S_3, whose start wavelength λ_1 is just before the start wavelength of S_4, has used up all the wavelengths between these two start wavelengths, it will use wavelengths near the start wavelength of S_4 to transmit bursts, so big collision may happen. Also in Figure 4.11 we find PWA-λ-TE suffers with the increase of W, actually it is worse than PWA-λ. This is due to the incompatibility of TE by path consideration and priorities without taking any path or link information into consideration. Finally, we note that PWA-TE and PWA-Link-TE are the two schemes that show a consistent drop in burst drop probability for the range of wavelengths considered here. Since these are the best performing schemes overall, this

routing algorithms are currently under investigation by our group.

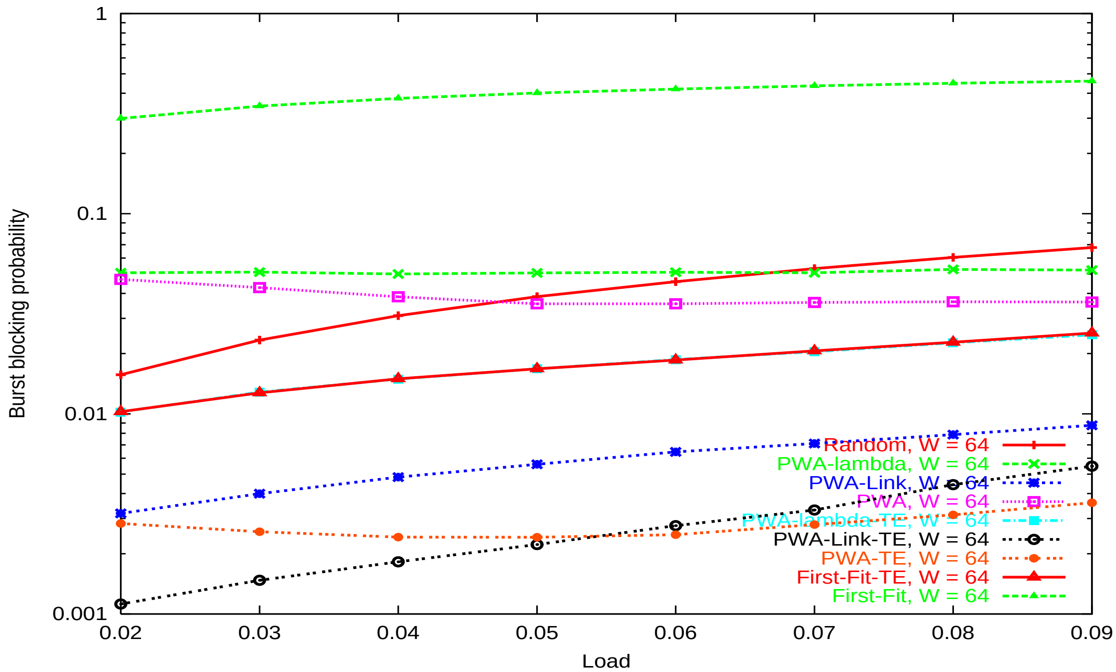

Figure 4.8: Burst drop probability, NSFNet, low load

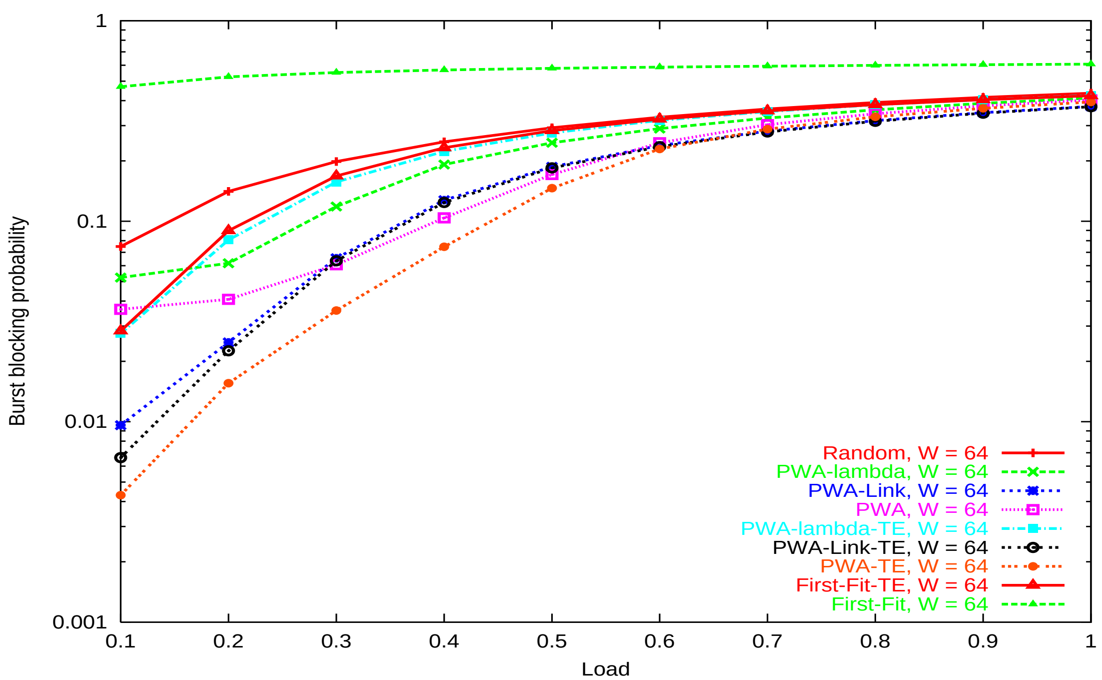

Figure 4.9: Burst drop probability, NSFNet, moderate and high load

result indicates that a combination of adaptive policies with traffic engineering is the best approach to take advantage of wavelength resources in the OBS network.

Figures 4.12 and 4.13 plot the burst drop probability as a function of the number of hops in a burst's path, for the torus and NSFNet topologies, respectively. As expected, the burst drop probability increases with the length of the path. However, while for some schemes (e.g., Random and First-Fit) there can be a difference of two orders of magnitude between the drop probability of bursts traversing one hop versus bursts that travel four hops, the difference is less acute when schemes employing adaptive policies with traffic engineering are used. Therefore, our approach not only improves the overall burst drop probability, it also increases the fairness among bursts. Again, we find that PWA-TE is the best performing scheme even when path lengths are taken into consideration.

Overall, the simulation results indicate that adaptive policies perform better than non-adaptive ones; and that applying traffic engineering techniques to achieve traffic isolation can further improve the performance of an OBS network in terms of burst drop probability and fairness. The PWA-TE scheme has been shown to perform the best over all the experiments we have conducted, with PWA-Link-TE a close second. Since PWA-TE is relatively easier to implement and involves fewer computations and memory requirements, it is the best choice for OBS networks with no wavelength conversion capabilities.

4.5 Concluding Remarks

We studied the wavelength assignment problem in OBS networks, and proposed a suite of policies based on the concepts of adaptivity and traffic engineering to achieve low burst blocking probability and to attain fairness among bursts with different path lengths. We believe it is possible to employ similar traffic engineering concepts to further reduce traffic interference in an OBS network, in particular by designing appropriate routing algorithms.

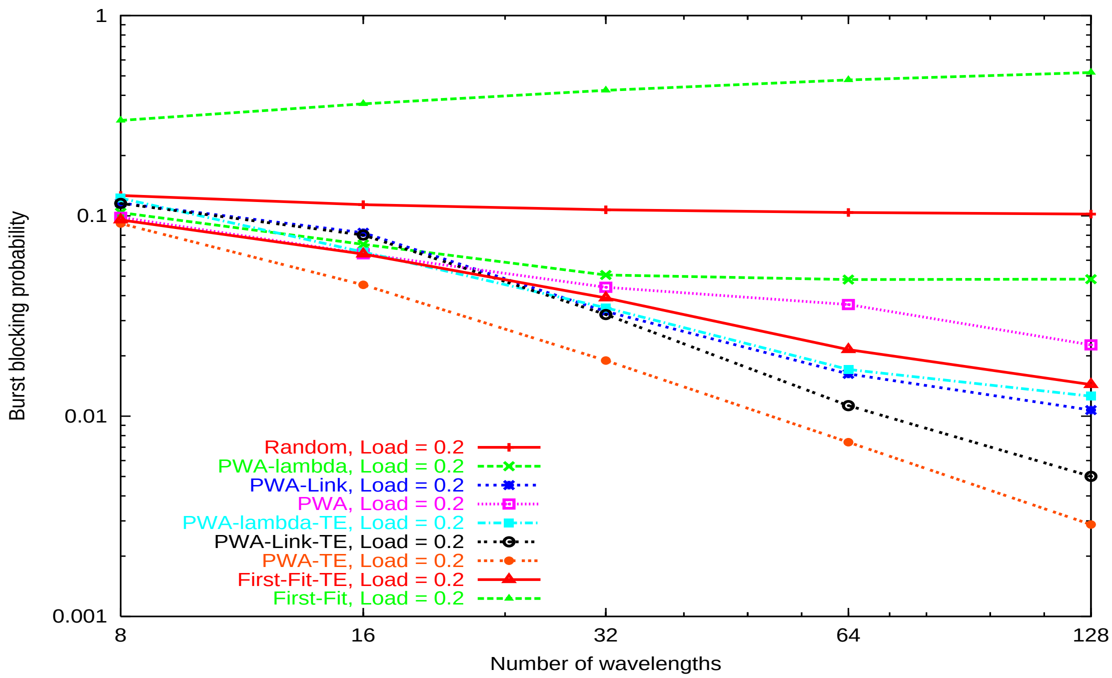

Figure 4.10: Burst blocking probability, 4×4 torus network, load $= 0.2$

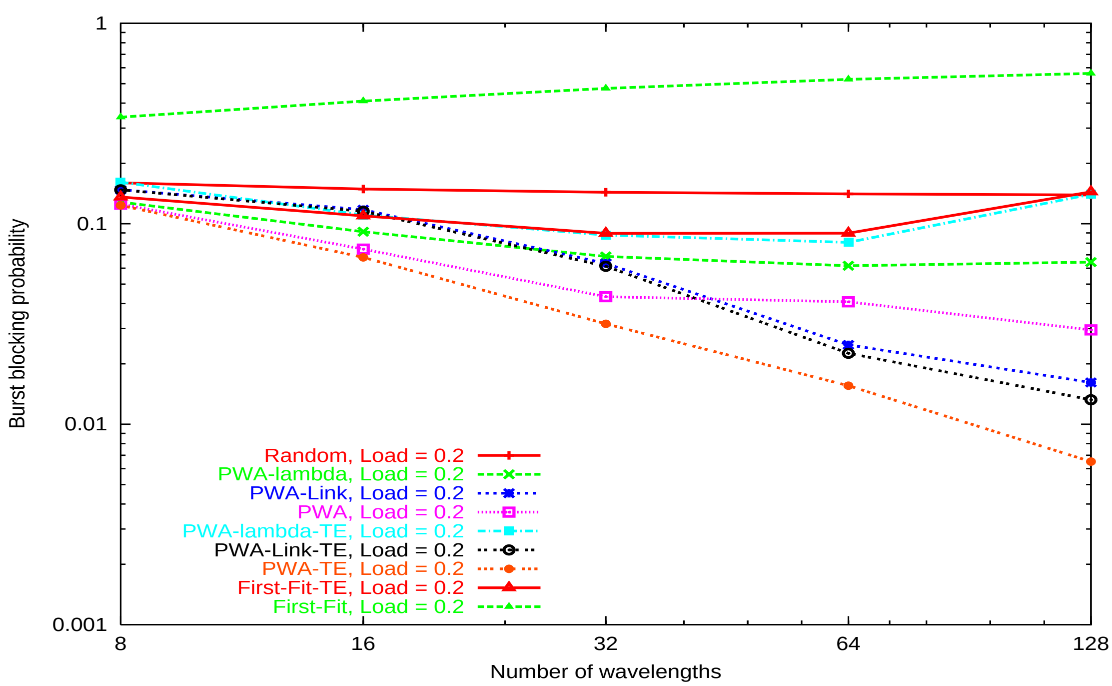

Figure 4.11: Burst blocking probability, NSFNet, load $= 0.2$

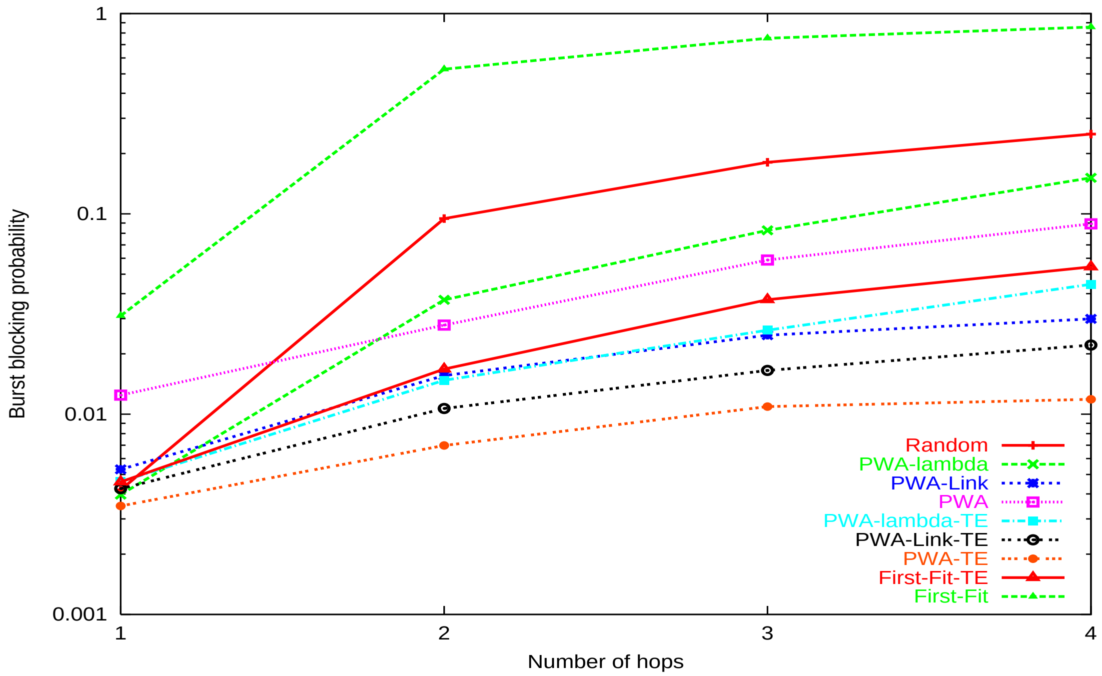

Figure 4.12: Burst blocking probability, 4×4 torus network, load $= 0.2$, W $= 64$

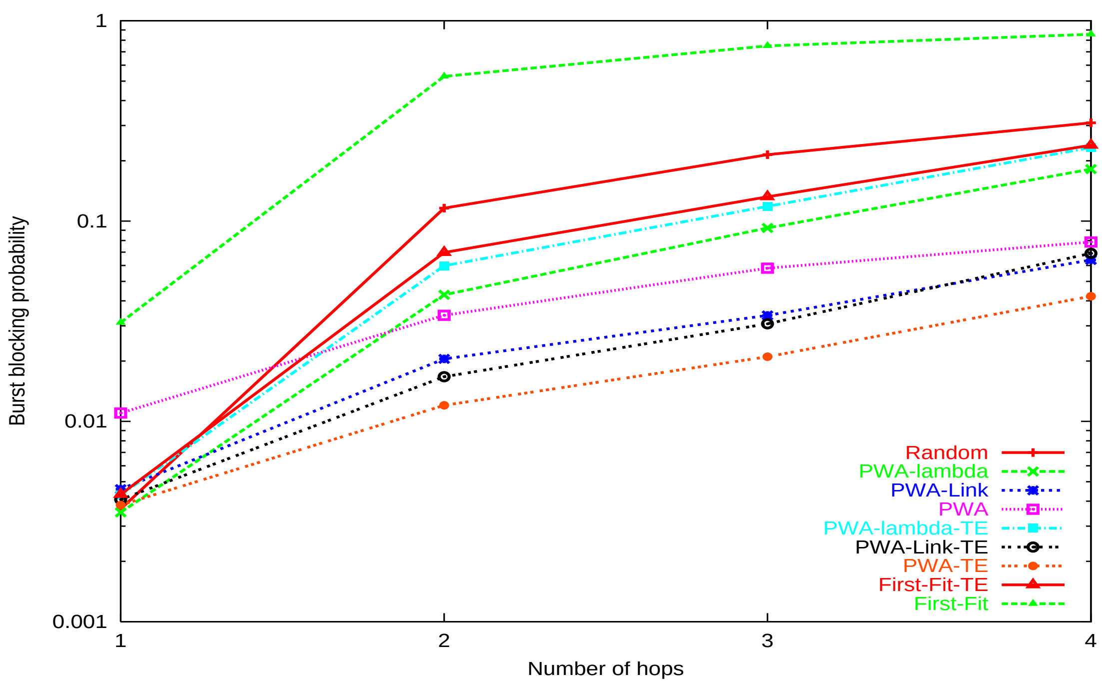

Figure 4.13: Burst blocking probability, NSF network, load $= 0.2$, W $= 64$

Chapter 5

Summary and Future Research

This thesis studies the optical burst switched networks. It has mainly three parts. First it presents a detailed analytical model for the JIT, JET, and Horizon wavelength reservation schemes for OBS networks. Though there has already been research on this subject, the models proposed in this thesis are more accurate, and valid for general burst length and offset length distributions. The thesis also proposes a new reservation scheme, JIT^+, which is as simple to implement as JIT, but its performance tracks that of Horizon and JET. From the results of both theoretical analysis and simulation, we find that, under reasonable assumptions regarding the system parameters of current and future state-of-the-art technologies in optical switch and electronic hardware, the performance of the simple schemes such as JIT and JIT^+ seems to get close to that of more complicated Horizon and JET.

Secondly this thesis presents the results of a simulation model of OBS networks employing the `Jumpstart` JIT signaling protocol. The performance (in terms of burst drop probability) of various network topologies is studied, and the effects of several system parameters are investigated, including message processing time, OXC configuration delay, user-to-switch propagation delay, and switch-to-switch propagation delay. We also investigate the effect of wavelength converters.

In the third part of the thesis, we develop a suite of adaptive and non-adaptive wavelength assignment policies for OBS switches. Through the study of static traffic pattern, traffic engineering techniques are used to reduce wavelength contention through traffic isolation. We also use adaptive wavelength assignment to makes different edge switches assign wavelengths to bursts by different orders to reduce burst confliction further. Our performance study indicates that, in the absence of full wavelength conversion capabilities, intelligent choices in assigning wavelengths to bursts at the source can significantly improve the burst drop probability in an OBS network.

5.1 Future Work

Our work can be extended in the following directions:

- Routing algorithms for OBS networks that take into account optical layer constraints to guarantee optical QoS. This is important because in OBS, bursts pass intermediate switches transparently, accumulating impairments along the path so that the quality of signal may suffer at the destination. In general, constrainted shortest path first (CSPF) algorithm, which is derived from Dijkstra's shortest path first algorithm, can be used to compute best path that satisfies the constraints required by the system [16]. However, this problem is NP-hard, therefore it is important to study this issue in depth and develop approximation algorithms or heuristics that are practical and efficient.

- Routing is one of the most important issues to be addressed to achieve the best performance in a network. Throughout this thesis, we have assumed that shortest path routing is used. However, shortest path routing may result in significant path overlap, leading to high burst blocking. We plan to develop new alternative routing algorithms that are more appropriate for burst switched networks to lower the burst blocking probability by balancing the traffic among different nodes and links. As a first step we would optimize routing in an OBS network with full wavelength conversion. Such a study could then be extended to combine optimized routing with the wavelength assignment policies in Chapter 4, to achieve better performance when wavelength conversion is not available.

- Placement of wavelength converters in an OBS network to minimize burst loss. If all the switches have full wavelength conversion capability, we can gain the best performance, however, at present, wavelength converters are very expensive. We will develop algorithms to place a limited number of wavelength converters inside an OBS network to minimize the burst loss. Furthermore, we would like to consider the placement of wavelength converters together with the selection of routing and wavelength assignment policies, as described above.

Appendix A

Estimation of the Parameter Δ for Horizon

We now consider the problem of estimating the value of parameter Δ in the expression (2.4) 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 Figure 2.4 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 Figure 2.4. This interval corresponds to the time until the next burst arrival, since any burst arriving after time t_5 is accepted. We let $Prob^{noburst}(u)$ 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 Figure 2.4, 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 [27], 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:

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

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:

$$Prob^{noburst}(u) \quad = \quad \int_{s=0}^{\infty}\int_{t=0}^{\infty}\frac{\lambda}{W}e^{-\lambda\left[t-\int_0^t G(s-x)dx\right]/W}\left[1-G(s+u-t)\right]dt\ dH(s) \quad \text{(A.2)}$$

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

$$\overline{u} \;=\; \int_0^{\infty} u dP(u) \;=\; \int_0^{\infty}(1-P(u))du \qquad \text{(A.3)}$$

Given the CDF $G(z)$ and $H(s)$, it is possible to compute $\overline{u}$ numerically. We then let $\Delta = \frac{\bar{u}}{W} - \frac{1}{\lambda}$ in the expression (2.4) for the traffic intensity of Horizon.

Bibliography

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

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

[3] Proceedings of the workshops on optical burst switching. http://www.cs.buffalo.edu/q̃iao/wobs/, 2003.

[4] 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*, 2003.

[5] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson. `JumpStart`: A just-in-time signaling architecture for WDM burst-switched networks. *IEEE Communications*, 40(2):82–89, February 2002.

[6] 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.

[7] Ayan Banerjee, John Drake, Jonathan P. Lang, Brad Turner, Kireeti Kompella, and Yakov Rekhter. Generalized multiprotocol label switching: An overview of routing and management enhancements. *IEEE Communications Magazine*, 39(1):144–150, January 2001.

[8] Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of ACM*, 22(2):248–260, 1975.

[9] Arnold Bragg, Ilia Baldine, and Dan Stevenson. A transport layer architectural framework for optical burst switched (obs) networks. In *Proceedings of the First International Workshop on Optical Burst Switching*, October 2003.

[10] F. Callegati, H. C. Cankaya, Y. Xiong, and M. Vandenhoute. Design issues of optical IP routers for internet backbone applications. *IEEE Communications*, pages 124–128, December 1999.

[11] X. Cao, J. Li, Y. Chen, and C. Qiao. Tcp/ip packets assembly over optical burst switching network. In *Proceeding of Globecom*, 2002.

[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] Angela L. Chiu and John Strand. An agile optical layer restoration method for router failures. *IEEE Network*, 17(2):38–42, March/April 2003.

[14] P. B. Chu, S-S. Lee, and S. Park. MEMS: The path to large optical crossconnects. *IEEE Communications*, 40(3):80–87, March 2002.

[15] Ralph B. D'Agostino and Michael A. Stephens. *Goodness-of-Fit Techniques*. Marcel Dekker, INC, 1986.

[16] B. Davie and Y. Rekhter. *MPLS Technology and Applications*. Morgan Kaufmann Publishers, San Diego, California, 2000.

[17] 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, 2002.

[18] K. Dolzer and C. Gauger. On burst assembly in optical burst switching networks - a performance evaluation of Just-Enough-Time. In *Proceedings of the 17th International Teletraffic Congress*, pages 149–161, September 2001.

[19] 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.

[20] M. Duser and P. Bayvel. Analysis of a dynamically wavelength-routed, optical burst switched network architecture. *IEEE/OSA Journal of Lightwave Technology*, 20(4):574–585, April 2002.

[21] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., New York, 1979.

[22] A. Ge, F. Callegati, and L. S. Tamil. On optical burst switching and self-similar traffic. *IEEE Communications Letters*, 4(3):98–100, March 2000.

[23] Andre Girard. *Routing and Dimensioning in Circuit-Switched Networks.* Addison Wesley Publishing Company, 1990.

[24] David Griffith and SuKyoung Lee. A 1+1 protection architecture for optical burst switched networks. *IEEE Journal on Selected Areas in Communications*, 21(9):1384–1398, November 2003.

[25] C. Hsu, T. Liu, and N. Huang. Performance analysis of deflection routing in optical burst-switched networks. In *Proceedings of INFOCOMM*, volume 1, pages 66–73, 2002.

[26] Paola Iovanna, Roberto Sabella, and Marina Settembre. A traffic engineering system for multilayer networks based on the gmpls paradigm. *IEEE Network*, 17(2):28–37, March/April 2003.

[27] L. Kleinrock. *Queueing Systems, Volume 1: Theory.* John Wiley & Sons, New York, 1975.

[28] E. Kozlovski and P. Bayvel. Link failure restoration in wavelength-routed optical burst switched (wr-obs) networks. In *Optical Fiber Communications Conference*, 2003.

[29] Eugene Kozlovski. Survivability of wavelength-routed optical burst-switched networks with guaranteed ip services. In *Proceedings of 2003 5th International Conference on Transparent Optical Networks*, 2003.

[30] J. Li and C. Qiao. Schedule burst proactively for optical burst switching networks. *Computer Networks*, 2004. To appear.

[31] J. Liu and N. Ansari. Forward resource reservation for qos provisioning in obs systems. In *Proceedings of Globecom*, volume 3, pages 17–21, 2002.

[32] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[33] 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.

[34] B. Ramamurthy and B. Mukherjee. Wavelength conversion in WDM networking. *IEEE Journal Selected Areas in Communications*, 16(7):1061–1073, September 1998.

[35] Rajiv Ramaswami and Kumar N. Sivarajan. *Optical Networks, A Practical Perspective*. Mogan Kaufmann Publishers, 1998.

[36] Zvi Rosberg, Hai Le Vu, Moshe Zukerman, and Jolyon White. Performance analysis of optical burst-switching networks. *IEEE Journal of Selected Area in Communications*, 21(7):1187–1197, Sep. 2003.

[37] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. RFC 3031, January 2001.

[38] L.B. Sofman, K. Laevens, and T.S. El-Bawab. Segmentation overhead in optical burst switching. In *Proceeding of Opticomm*, pages 101–108, 2002.

[39] J. S. Turner. Terabit burst switching. *Journal of High Speed Networks*, 8(1):3–16, January 1999.

[40] 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.

[41] V. Vokkarane, J.P.Jue, and S. Sitaraman. Burst segmentation: An approach for reducing packet loss in optical burst switched networks. In *Proceedings of IEEE ICC*, volume 5, pages 2673–2677, 2002.

[42] X. Wang, H. Morikawa, and T. Aoyama. Priority-based wavelength assignment algorithm for burst switched WDM optical networks. *IEICE Transactions on Communications*, E86-B(5):1508–1514, May 2003.

[43] 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.

[44] 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.

[45] Xipeng Xiao, Alan Hannan, Brook Bailey, and Lionel M. Ni. Traffic engineering with mpls in the internet. *IEEE Network*, 14(2):28–33, March/April 2000.

[46] Y. Xiong, M. Vandenhoute, 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.

[47] J. Xu, C. Qiao, J. Li, and G. Xu. Efficient channel scheduling algorithms in optical burst switched networks. In *Proceedings of IEEE INFOCOM*, 2003.

[48] L. Xu, H. G. Perros, and G. N. Rouskas. Techniques for optical packet switching and optical burst switching. *IEEE Communications*, 39(1):136–142, January 2001.

[49] L. Xu, H. G. Perros, and G. N. Rouskas. Performance analysis of an edge optical burst switching node with a large number of wavelengths. In *Proceedings of the Eighteenth International Teletraffic Congress (ITC 18)*, pages 891–900. Elsevier Science, September 2003.

[50] 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.

[51] L. Xu, H. G. Perros, and George N. Rouskas. A simulation study of optical burst switching access protocols for WDM ring networks. *Computer Networks*, 41(2):143–160, January 2003.

[52] X.Yu, Y.Chen, and C.Qiao. Performance evaluation of optical burst switching with assembled burst traffic input. *Proceedings of IEEE Globecom*, pages 2318–2322, 2002.

[53] X.Yu, Y.Chen, and C.Qiao. A study of traffic statistics of assembled burst traffic in optical burst switched networks. *Proceedings of Opticomm*, pages 149–159, 2002.

[54] M. Yang, S. Q. Zheng, and D. Verchere. A QoS supporting scheduling algorithm for optical burst switching DWDM networks. In *Global Telecommunications Conference*, pages 86–91, 2001.

[55] M. Yoo and C. Qiao. Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks. In *IEEE/LEOS Technol. Global Information Infrastructure*, pages 26–27, August 1997.

[56] 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.

[57] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, and D. Stevenson. Formal description of the `JumpStart` just-in-time signaling protocol using EFSM. In *Opticomm 2002*, pages 160–173, July 2002.

[58] 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.

[59] Andrew Zalesky, Hai Le Vu, Zvi Rosberg, Eric W. M. Wong, and Moshe Zukerman. Reduced load erlang fixed point analysis of optical burst switched networks with deflection routing and wavelength reservation. In *Proceedings of the First International Workshop on Optical Burst Switching*, Oct. 2003.

[60] Keyao Zhu, Hongyue Zhu, and Biswanath Mukherjee. Traffic engineering in multigranularity heterogeneous optical wdm mesh networks through dynamic traffic grooming. *IEEE Network*, 17(2):8–15, March/April 2003.

[61] Y. Zhu, G. N. Rouskas, and H. G. Perros. A comparison of allocation policies in wavelength routing networks. *Photonic Network Communications*, 2(3):265–293, August 2000.