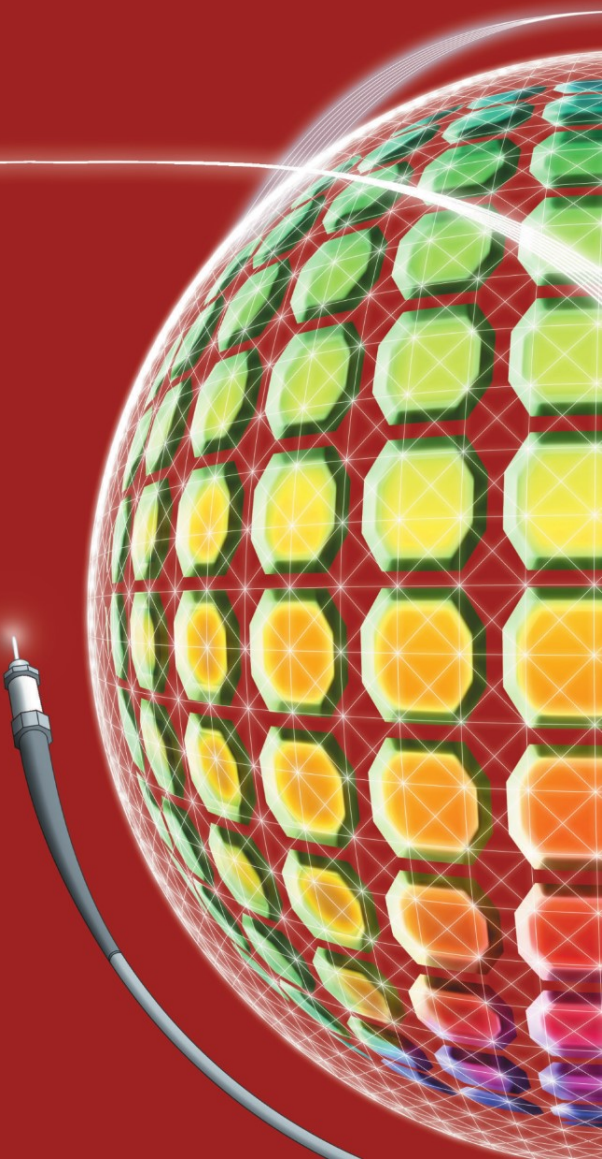
Optical Burst Switched Networks

Jason P. Jue
Vinod M. Vokkarane



OPTICAL NETWORKS SERIES

Optical Burst Switched Networks

OPTICAL NETWORKS SERIES

Series Editor
Biswanath Mukherjee, *University of California, Davis*

OPTICAL BURST SWITCHED NETWORKS

JASON P. JUE
The University of Texas at Dallas

VINOD M. VOKKARANE
University of Massachussetts Dartmouth

Jason P. Jue
The University of Texas at Dallas
Dept. of Computer Science
P.O. Box 830688
Richardson, TX 75083-0688

Vinod M. Vokkarane
University of Massachusetts, Dartmouth
Dept. of Computer & Information Science
285 Old Westport Road
North Dartmouth, MA 02747-2300

Optical Burst Switched Networks

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

ISBN 0-387-23756-9 e-ISBN 0-387-23760-7 Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11054542

springeronline.com

To the memory of my
brother, Jeff
— Jason P. Jue

To my parents
— Vinod M. Vokkarane

Contents

List of Figures

List of Tables

Preface

The amount of research being done in the area of optical burst switching has increased tremendously in the past several years. A few years ago, there were very few research works on optical burst switching. Now, entire technical sessions at major conferences are being devoted to the topic of optical burst switching, and several workshops on optical burst switching have been organized. An Optical Burst Switching Forum has been formed to facilitate research in optical burst switching through discussion and collaboration, and several international efforts are under way to develop optical burst-switched networks. The amount of research and development being devoted to optical burst switching is a good indication of the significant potential of optical burst-switched networks. Optical burst-switched networks have the potential to provide flexible all-optical data transmission without significant technological barriers.

The book is intended to provide an overview of optical burst switching. Since the amount of research in optical burst switching is growing rapidly, it is impossible to cover every research work on the topic. Instead, we attempt to identify key areas in optical burst switching, and outline the various design choices and parameters in each of these areas. We then discuss several research papers within the context of this general framework and present in-depth coverage of selected scheme and techniques.

The book is organized into eight chapers. Chapter 1 provides an introduction to optical burst switching, with comparisons to optical circuit and packet switching techniques. Chapter 2 presents an overview of the optical burst switching node architecture and discusses several component-level and physical-layer issues in optical burst-switched networks. Chapter 3 discusses the issue of burst assembly at the edge nodes. Chapter 4 discusses several signaling schemes for reserving resources in an optical burst-switched network. Chapter 5 discusses the is-

sue of contention in optical burst-switched networks and presents several approaches for resolving contention. Chapter 6 discusses the problem of scheduling bursts on wavelength-division multiplexed links. Chapter 7 presents an overview of quality of service schemes for optical burst-switched networks. Finally, Chapter 8 discusses various additional issues in optical burst-switched networks, such as survivability, mulitcasting, and the interaction of optical burst switching with higher-layer protocols and applications.

Much of this book is the result of research that we have conducted with graduate students in the Advanced Networks Research Lab at the University of Texas at Dallas, and we would like to acknowledge the contributions of these students. In particular, we would like to thank Qiong Zhang for her contributions in the area of quality of service in optical burst-switched networks, Farid Farahmand, Tao Zhang, and Guru Thodime for their contributions related to contention resolution, Ravikiran Karanam for his contributions in signaling protocols, Karthik Haridoss for his work on burst assembly, and Sriranjani Sitaraman for her work on burst segmentation.

We would also like to acknowledge support from the National Science Foundation (NSF) through grant ANI-01-33899. Without this funding, the book would not have been possible. Finally, we would like to thank our parents for their love, support, and encouragement over the years.

Jason P. Jue, *jjue@utdallas.edu*
Vinod M. Vokkarane, *vvokkarane@umassd.edu*
September 2004

Chapter 1

INTRODUCTION

Over the last decade, the field of networking has experienced growth at a tremendous rate. The rapid expansion of the Internet and the ever-increasing demand for multimedia information are severely testing the limits of our current computer and telecommunication networks. There is an immediate need for the development of new high-capacity networks that are capable of supporting these growing bandwidth requirements.

In order to meet these growing needs, optical wavelength-division multiplexing (WDM) communication systems have been deployed in many telecommunications backbone networks. In WDM systems, each fiber carries multiple communication channels, with each channel operating on a different wavelength. Such optical transmission systems have the potential to provide over 50 Tb/s on a single fiber.

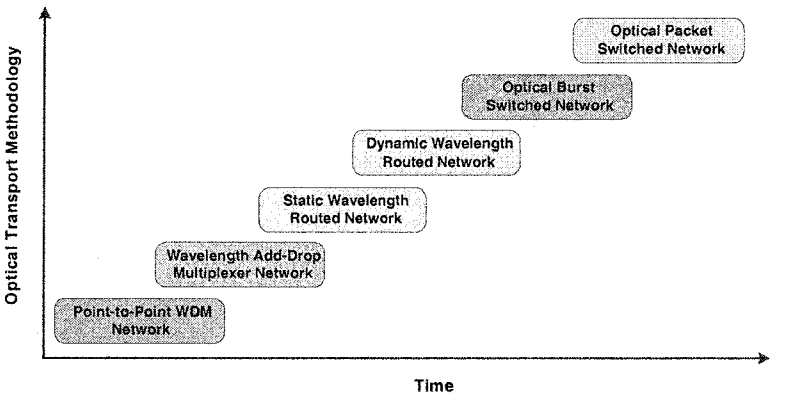


Figure 1.1. Evolution of optical transport methodologies.

Figure 1.1 shows the evolution of the different optical transport methodologies [1]. The first generation optical network architectures consist of *point-to-point WDM links.* Such networks are comprised of several point-to-point links at which all traffic arriving to a node is dropped, converted from optics to electronics, processed electronically, and converted from electronics to optics before departing from the node. The dropping and adding of traffic at every node in the network incurs significant overhead in terms of switch complexity and electronic processing cost, particularly if the majority of the traffic in the network happens to be bypass traffic. In order to minimize the network cost, all-optical devices can be used.

Second-generation optical network architectures are based on wavelength add-drop multiplexers (WADM) [2], where traffic can be added and dropped at WADM locations. WADMs can allow selected wavelength channels on a fiber to be terminated, while other wavelengths pass through untouched. In general, the amount of bypass traffic in the network is significantly higher than the amount of traffic that needs to be dropped at a specific node; hence, by using WADMs, we can reduce the overall network cost. WADMs are primarily used to build optical WDM ring networks, which are expected to deployed in metropolitan-area markets.

In order to build a mesh network consisting of multi-wavelength fiber links, appropriate fiber interconnection devices are needed. Third-generation optical network architectures are based on all-optical interconnection devices. These devices fall under three broad categories, namely *passive star couplers*, *passive routers*, and *active switches* [3]. The *passive star* is a broadcast device. A signal arriving on a given wavelength on input fiber port of the star coupler will have its power equally divided among all output ports of the star coupler. A *passive router* can separately route each of several wavelengths arriving on an input fiber to the same wavelength on different output fibers. The passive router is a static device; thus, the routing configuration is fixed. An *active switch* also routes wavelengths from input fibers to output fibers and can support simultaneous connections. Unlike a passive router, the active switch can be reconfigured to change the interconnection pattern of incoming and outgoing wavelengths. In these third-generation optical networks, data is allowed to bypass intermediate nodes without undergoing conversion to electronics, thereby reducing the costs associated with providing high-capacity electronic switching and routing capabilities at each node.

Emerging all-optical systems are expected to provide optical circuit-switched connections, or lightpaths [23], between edge routers over an optical core network; however, since these circuit-switched connections

are fairly static, they may not be able to accommodate the bursty nature of Internet traffic in an efficient manner. Ideally, in order to provide the highest possible utilization in the optical core, nodes would need to provide packet switching at the optical level [12, 6]. Such all-optical packet switching is likely to be infeasible in the near future due to technological constraints.

A possible near-term alternative to all-optical circuit switching and all-optical packet switching is *optical burst switching* [6]. In optical burst switching, packets are concatenated into transport units referred to as bursts. The bursts are then switched through the optical core network in an all-optical manner. Optical burst-switched networks allow for a greater degree of statistical multiplexing and are better suited for handling bursty traffic than optical circuit-switched networks. At the same time, optical burst-switched networks do not have as many technological constraints as all-optical packet-switched networks. In order to highlight the differences between optical circuit switching, optical packet switching, and optical burst switching, we discuss each approach in detail.

1.1 Optical Circuit Switching

Wavelength-routed optical networks employ optical circuit switching in which all-optical wavelength paths (lightpaths) are established between pairs of nodes. The establishment of lightpaths involves several tasks. These tasks include topology and resource discovery, routing, wavelength assignment, and signaling and resource reservation.

Topology and resource discovery involves the distribution and maintenance of network state information. Typically this information will include information on the physical network topology and the status of links in the network. In a wavelength-routed WDM network, this information may include the availability of wavelengths on a given link in the network. A common protocol for maintaining link state information in the Internet is the Open Shortest Path First (OSPF) protocol.

The problem of finding routes and assigning wavelengths for lightpaths is referred to as the routing and wavelength assignment (RWA) problem. Typically, connection requests may be of two types, *static and dynamic.* In the *Static Lightpath Establishment* (SLE) problem, the entire set of connections is known in advance, and the problem is to set up lightpaths for these connections while minimizing network resources such as the number of wavelengths or the number of fibers in the network. For the *Dynamic Lightpath Establishment* (DLE) problem, a lightpath is set up for each connection request as it arrives, and the lightpath is released after some finite amount of time. The objective in the dynamic traffic cases is to set up lightpaths and assign wavelengths in a manner

which minimizes the amount of connection blocking or which maximizes the number of connections that are established in the network at any time. There has been extensive research to solve both the static and the dynamic RWA problems [19].

Wavelength-routed lightpath connections are fairly static and may not be able to accommodate the highly variable and bursty nature of Internet traffic in an efficient manner. It is clear that if traffic is varying dynamically, then sending this traffic over static lightpaths would result in the inefficient utilization of bandwidth. On the other hand, if we attempt to set up lightpaths in a very dynamic manner, then the network state information will be constantly changing, making it difficult to maintain current network state information. Thus, as traffic becomes more dynamic and bursty in nature, alternative approaches may be needed to transport data across networks.

1.2 Optical Packet Switching

As optical switching technology improves, we may eventually see the emergence of photonic packet-switched networks in which packets are switched and routed independently through the network entirely in the optical domain without conversion back to electronics at each node. Such photonic packet-switched networks allow for a greater degree of statistical multiplexing on optical fiber links and are better suited for handling bursty traffic than optical circuit-switched networks.

An example of a basic photonic packet-switch architecture is shown in Fig. 1.2. A node contains an optical switch fabric which is capable of reconfiguration on a packet-by-packet basis. The switch fabric is reconfigured based on information contained within the header of a packet. The header itself is typically processed electronically, and can either be carried in-band with the packet, carried on a subcarrier frequency, or carried out-of-band on a separate control channel. Since it takes some time for the header to be processed and for the switch to be reconfigured, the packet may be delayed by sending it through an optical delay line.

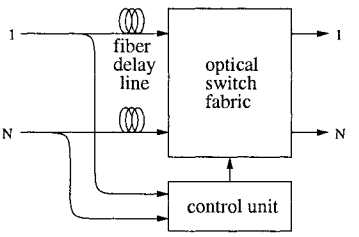


Figure 1.2. A photonic packet-switch architecture.

In order for photonic packet switching to be practical, fast switching times are required. Currently, switching times for MEM-based switches are on the order of 1 to 10 ms, while semiconductor optical amplifier-based switches have switching times which are less than 1 ns [6]. The disadvantage of semiconductor optical amplifier switches is that they tend to be more expensive, and the switch architectures require signals to pass through optical couplers which results in additional power losses. While switching speeds are expected to improve in the near future, current technology is not yet mature enough to support photonic packet switching.

Another challenge in photonic packet switching is synchronization. In photonic packet-switched networks with fixed-length packets, synchronization of packets at switch input ports is often desired in order to minimize contention. Although synchronization is typically difficult to achieve, a few synchronization techniques have been proposed and implemented in laboratory settings [10, 11].

Since network resources are not reserved in advance in photonic packet switching, packets may experience contention in the network. Contention occurs when two or more packets contend for the same output port at the same time. Typically, contention in traditional electronic packet-switching networks is handled through buffering; however, in the optical domain, it is more difficult to implement buffers, since there is no optical equivalent of random-access memory. Instead, optical buffering is achieved through the use of fiber delay lines [3, 4]. By implementing multiple delay lines in stages [3] or in parallel [4], a buffer may be created which can hold a packet for a variable amount of time. Some papers have investigated approaches for designing larger buffers without a large number of delay lines [6, 7]. In [6], the buffer size is increased by cascading multiple stages of delay lines. In [7], the buffer size is increased by utilizing so called non-degenerate buffers in which the length of the delay lines may be greater than the number of delay lines in the buffer. This approach yields lower packet loss probabilities, but does not guarantee the correct ordering of the packets. Note that, in any optical buffer architecture, the size of the buffers is severely limited, not only by signal quality concerns, but also by physical space limitations. To delay a single packet for 5 μs requires over a kilometer of fiber. Because of this size limitation of optical buffers, a node may be unable to effectively handle high load or bursty traffic conditions.

Another approach to resolving contention is to route the contending packets to an output port other than the intended output port. This approach is referred to as deflection routing or hot-potato routing [21–23]. While deflection routing is generally not favored in electronic

packet-switched networks due to potential looping and out-of-sequence delivery of packets, it may be necessary to implement deflection in photonic packet-switched networks, where buffer capacity is very limited, in order to maintain a reasonable level of packet losses. However, before attempting to deploy deflection in photonic packet-switched networks, a comprehensive study is required in order to identify potential methods for overcoming some of the limitations of deflection, and to determine whether or not these methods, along with the potential benefits of deflection, are sufficient to justify implementation.

1.3 Optical Burst Switching

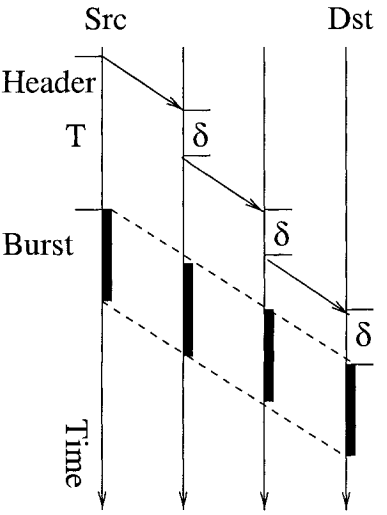


Figure 1.3. The use of offset time in OBS.

Optical burst switching is designed to achieve a balance between optical circuit switching and optical packet switching. In an optical burst-switched network, a data burst consisting of multiple IP packets is switched through the network all-optically. A control packet is transmitted ahead of the burst in order to configure the switches along the burst's route. The offset time (Figure 1.3) allows for the control packet to be processed and the switch to be set up before the burst arrives at the intermediate node; thus, no electronic or optical buffering is necessary at the intermediate nodes while the control packet is being processed. The control packet may also specifies the duration of the burst in order to let the node know when it may reconfigure its switch for the next arriving burst.

Optical Switching Paradigm	Bandwidth Utilization	Setup Latency	Switching SpeedReq.	Proc. / Sync. Overhead	Traffic Adaptively
Optical Circuit Switching	Low	High	Slow	Low	Low
Optical Packet Switching	High	Low	Fast	High	High
Optical Burst Switching	High	Low	Medium	Low	High

Figure 1.4. Comparison of the different all-optical network technologies.

By reserving resources only for a specified period of time rather than reserving resources for an indefinite period of time, the resources can be allocated in a more efficient manner and a higher degree of statistical multiplexing can be achieved. Thus, optical burst switching is able to overcome some of the limitations of static bandwidth allocation incurred by optical circuit switching. Furthermore, since data is transmitted in large bursts, optical burst switching reduces the technological requirement of fast optical switches that is necessary for optical packet switching.

Figure 1.4 summarizes the three different all-optical transport paradigms. From the figure, we can clearly observe that optical burst switching has the advantages of both optical circuit switching (or wavelength routed networks) and optical packet switching, while potentially avoiding their shortcomings.

Although optical burst switching appears to offer advantages over optical circuit switching and optical packet switching, several issues need to be considered before optical burst switching can be deployed in working networks. In particular, these issues include burst assembly, signaling schemes, contention resolution, burst scheduling, and quality of service.

A burst assembly scheme is required to determine how packets are assembled into bursts. Issues include when to assemble a burst, how many packets to include in a burst, and what types of packets to include in a burst. The burst assembly scheme will affect the burst length as well as the amount of time that a packet must wait before being transmitted. Assembly schemes based on timer and threshold mechanisms have been proposed in the literature and are discussed in Chapter 3.

A signaling scheme is required for reserving resources and configuring switches for an arriving burst. Common signaling schemes for reserving resources in OBS networks are tell-and-go (TAG), tell-and-wait (TAW), and just-enough-time (JET). In the TAG scheme, the source nodes sends out a control message to notify downstream nodes of a burst's arrival. The source node then immediately follows the control message with the data burst, without waiting for an acknowledgement [19, 20]. In order to allow time for the processing of the control message and the configuring of the switch at each node, the burst may need to be buffered at each node. In the TAW scheme [20], the source sends a control message to reserve resources for the burst along the path. The source then waits for an acknowledgement confirming that the reservations have been successful. Upon receiving a positive acknowledgement, the source will send the burst. Otherwise, the source will need to reattempt the reservation. In JET [6], there is a delay between transmission of the control packet and transmission of the optical burst. This delay can be set to be larger than the total processing time of the control packet along the path. Thus, when the burst arrives at each intermediate node, the control packet has been processed and a channel on the output port has been allocated. Therefore, there is no need to buffer the burst at the node. This is a very important feature of the JET scheme, since optical buffers are difficult to implement. A further improvement of the JET scheme can be obtained by reserving resources at the optical burst switch from the time the burst arrives at the switch, rather than from the time its control packet is processed at the switch. Different signaling techniques for OBS networks are studied in detail in Chapter 4.

In the TAG and JET schemes, the source does not wait for an acknowledgement before sending a burst. Thus, it is possible that the reservations will not be successful at some node in the path. In this case, a burst that is in transit will experience contention. Contention occurs when more than one burst contends for the same resource at the same time. Contention may be resolved in a number of ways. One approach is to store one of the bursts until the appropriate resources become available. Another approach is to deflect one of the bursts to a different output port. A third approach is to convert one of the bursts to a different wavelength on the output fiber. When contention resolution techniques are not successful, then a burst must be dropped. One approach to reduce the amount of data lost during a contention is *burst segmentation.* In burst segmentation, only those parts of a burst that overlap with another burst will be dropped. Contention resolution schemes and burst segmentation are discussed in Chapter 5.

When wavelength conversion is used, one problem is to determine the appropriate wavelength for a burst on an output link. This problem is referred to as *channel scheduling.* Several channel scheduling schemes that attempt to maximize channel utilization have been developed by researchers. These schemes are discussed in detail in Chapter 6.

A significant issue in networks is providing quality of service (QoS) for applications with varying requirements. Many burst assembly schemes, signaling protocols, contention resolution schemes, and channel scheduling schemes can be modified to provided differentiated services for different classes of traffic. Some of these approaches will be described in Chapter 7.

References

[1] B. Mukherjee. *Optical Communications Networks.* McGraw-Hill, New York, 1997.

[2] R.C. Alferness, H. Kogelnik, and T.H. Wood. The evolution of optical systems: Optics everywhere. *Bell Labs Technical Journal*, 5(1), Jan-March 2000.

[3] B.Mukherjee. WDM optical communication networks: Progress and challenges. *IEEE Journal on Selected Areas in Communications*, 18(10), OCTOBER 2000.

[4] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANs. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.

[5] S. Yao, B. Mukherjee, and S. Dixit. Advances in photonic packet switching: An overview. *IEEE Communications Magazine*, 38(2):84–94, February 2000.

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

[7] 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.

[8] H. Zang, J.P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *SPIE Optical Networks Magazine*, 1(1), January 2000.

[9] R. Ramaswami and K.N.Sivarajan. *Optical Networks: A Practical Perspective.* Morgan Kaufmann Publishers, 1998.

[10] D. J. Blumentahl, P. R. Prucnal, and J. R. Sauer. Photonic packet switches: Architectures and experimental implementation. *Proceedings of the IEEE*, 82(11):1650–1667, November 1994.

[11] M. C. Cardakli and A. E. Willner. Optical packet and bit synchronization of a switching node using fbg optical correlators. In *ofc*, March 2001.

[12] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and et al. CORD: Contention resolution by delay lines. *IEEE Journal on Selected Areas in Communications*, 14(5):1014–1029, June 1996.

[13] Z. Haas. The 'Staggering Switch': An electronically controlled optical packet switch. *IEEE/OSA Journal of Lightwave Technology*, 11(5/6):925–936, May/June 1993.

[14] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, and et al. SLOB: A switch with large optical buffers for packet switching. *IEEE/OSA Journal of Lightwave Technology*, 16(10):1725–1736, October 1998.

[15] L. Tancevski, G. Castanon, F. Callegati, and L. Tamil. Performance of an optical IP router using non-degenerate buffers. In *Proceedings, IEEE Globecom*, pages 1454–1459, December 1999.

[16] A. S. Acampora and I. A. Shah. Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing. *IEEE Transactions on Communications*, 40(6):1082–1090, June 1992.

[17] F. Forghieri, A. Bononi, and P. R. Prucnal. Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks. *IEEE Transactions on Communications*, 43(1):88–98, January 1995.

[18] A. Bononi, G. A. Castanon, and O. K. Tonguz. Analysis of hot-potato optical networks with wavelength conversion. *IEEE/OSA Journal of Lightwave Technology*, 17(4):525–534, April 1999.

[19] E. Varvarigos and V. Sharma. The ready-to-go virtual circuit protocol: A loss-free protocol for multigigabit networks using FIFO buffers. *IEEE/ACM Transactions on Networking*, 5:705–718, October 1997.

[20] I. Widjaja. Performance analysis of burst admission control protocols. *IEEE Proc. Commun.*, 142:7–14, February 1995.

Chapter 2

TECHNOLOGY AND ARCHITECTURE

The development of optical burst switching relies on the successful development of several key technologies, including all-optical switches, burst mode receivers, and optical wavelength converters. While development in these areas has progressed over the past several years, additional work may be required before such technology is available for use in practical systems. Regardless of what type of technology is eventually used in the design of optical burst-switched networks, network designers must stil take into consideration any physical-layer constraints imposed by the selected device and component technologies.

This chapter presents an architectural overview of optical burst switching nodes, focusing on the functional components needed for optical burst switching. We then present several key technologies for supporting the optical burst switching architecture and discuss various physical-layer issues that may affect the performance of optical burst-switched networks.

2.1 OBS Network Architecture

An optical burst-switched network consists of optical burst switching nodes that are interconnected via fiber links. Each fiber link capable of supporting multiple wavelength channels using wavelength division multiplexing (WDM). Nodes in an OBS network can either be *edge nodes* or *core nodes* as shown in Fig. 2.1. Edge nodes are responsible for assembling packets into bursts, and scheduling the bursts for transmission on outgoing wavelength channels. The core nodes are primarily responsible for switching bursts from input ports to output ports based on the burst header packets, and for handling burst contentions.

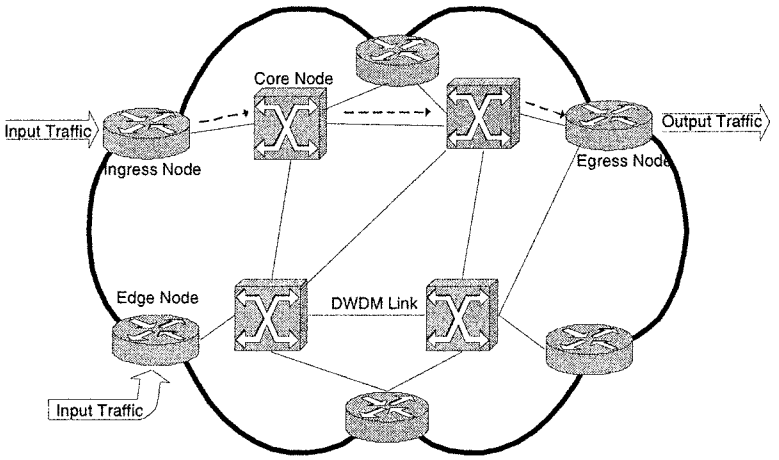


Figure 2.1. OBS Network Architecture

The ingress edge node assembles incoming packets from the client terminals into bursts. The assembled bursts are transmitted all-optically over OBS core routers without any storage at intermediate nodes within the core. The egress edge node, upon receiving the burst, disassembles the bursts into packets and forwards the packets to the destination client terminals. Basic architectures for core and edge routers in an OBS network have been studied in [13, 2, 3]. Figure 2.2, illustrates where various functionalities are implemented within an optical burst-switched network. The ingress edge node is responsible for burst assembly, routing, wavelength assignment, and scheduling of bursts at the edge. The core node is responsible for signaling, scheduling bursts on core links, and resolving contention. The egress edge node is primarily responsible for disassembling the burst and sending the packets up to the higher network layer.

In the network architecture, it can be assumed that each node can support both new input traffic as well as all-optical transit traffic. Hence, each node consists of both a core router and an edge router, as shown in Fig. 2.3 and Fig. 2.4.

The core routers (Fig. 2.3) consist of an optical cross connect (OXC) and a switch control unit (SCU). The SCU creates and maintains a forwarding table and is responsible for configuring the OXC [4]. When the SCU receives a burst header packet, it identifies the intended destination and consults the router signaling processor to find the intended output port. If the output port is available when the data burst arrives, the SCU configures the OXC to let the data burst pass through. If the port

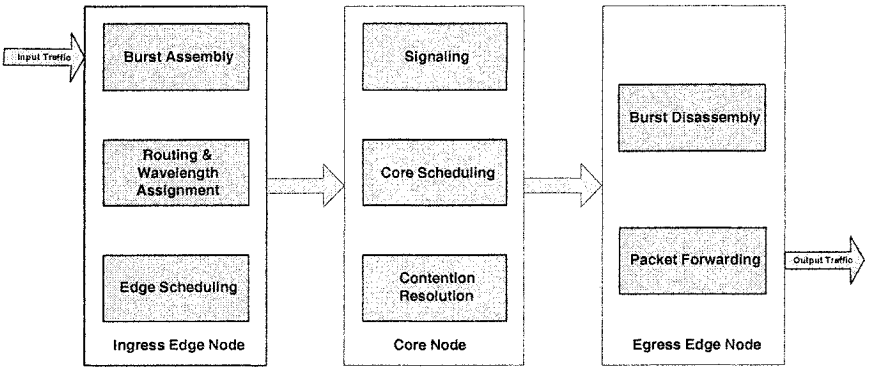


Figure 2.2. OBS functional diagram.

is not available, then the OXC is configured depending on the contention resolution policy implemented in the network. In general, the SCU is responsible for header interpretation, scheduling, collision detection and resolution, forwarding table lookup, switching matrix control, header rewrite, and wavelength conversion control. In the case of a data burst entering the OXC before its control packet, the burst is simply dropped (referred to as *early burst arrival problem*).

The edge router (Fig. 2.4) performs the functions of pre-sorting packets, buffering packets, assembling packets into burst, and disassembling bursts into its constituent packets. Different burst assembly policies, such as a threshold policy or a timer mechanism can be used to aggregate bursty data packets into optical bursts and to send the bursts into the network. The architecture of the edge router consists of a routing module (RM), a burst assembler, and a scheduler. The routing module selects the appropriate output port for each packet and sends each packet to the corresponding burst assembler module. Each burst assembler module assembles bursts consisting of packets which are headed for a specific egress router. In the burst assembler module, there is a separate packet queue for each class of traffic. The scheduler creates a burst based on the burst assembly technique and transmits the burst through the intended output port. At the egress router, a burst disassembly module disassembles the bursts into packets and send the packets to the upper network layers.

Some researchers have also proposed a more centralized OBS architecture, referred to as wavelength-routed optical burst switching (WR-OBS) [5]. A WR-OBS network combines the functions of OBS with fast circuit switching by dynamically assigning and releasing wavelength-routed lightpaths over a bufferless optical core. The potential advantages

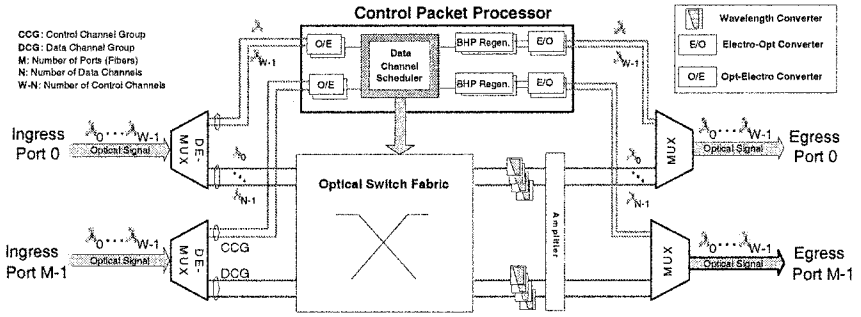


Figure 2.3. Architecture of Core Router.

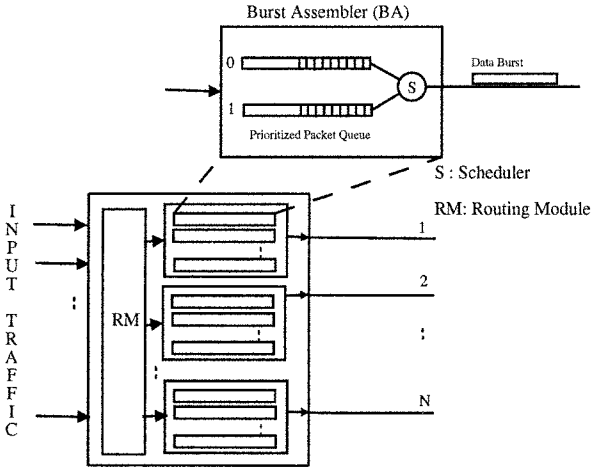


Figure 2.4. Architecture of Edge Router.

of this architecture compared to conventional OBS are explicit QoS provisioning. The benefits compared to static wavelength-routed optical networks (WRONs) are fast adaptation to dynamic traffic changes in optical networks and more efficient utilization of each wavelength channel.

In a WR-OBS network, a centralized request server is responsible for reserving resources for different connection request across the network. Each ingress node sends their connection request to the request server, where the requests are queued in based on their destination egress node and QoS class. The centralized server performs resource allocation based on its global knowledge of the status of every wavelength on every link in the entire network. The centralized request server is responsible for processing each individual connection request, calculating a route from

the source of the request to the corresponding destination, and also reserving the requested number of wavelengths on every link along the path of the connection. The ingress edge node begins data transmission only after it receives a confirmation message from the request server. WR-OBS may improve network throughput, but the centralized nature of the design is not very scalable.

2.2 Enabling Technology

In order to provide basic optical burst switching functionality described in the previous section, several optical device technologies are required. In core and edge nodes, the OXC must be implemented using a fast optical switch fabric. The edge nodes must also have fast burst-mode receivers that are able to acquire the signal of an incoming burst quickly. Each node should also have some form of wavelength conversion in order to reduce contention on output links.

2.2.1 Optical Switching Technology

While OBS does not require switching times as fast as optical packet switching, fast switching times are nonetheless favorable. Currently, there are several different candidate technologies for performing all-optical switching.

One of the more mature device technologies for performing all-optical switching is micro-electromechanical systems (MEMS) technology. In MEMS switches, tiny movable mirrors are adjusted to direct light from a given input port to a given output port. One example of how MEMS switches can be designed is given in Fig. 2.5. In this design, the light from a given input fiber is directed to a mirror in an input mirror array. The mirror is adjusted to redirect the light to a mirror in an output mirror array which directs the light to the appropriate output fiber. Since MEMS rely on mechanical adjustment of mirrors to redirect light, switching times are somewhat slow. Typical switching times for MEMS switches are on the order of 50 ms.

A switching technology that offers faster switching times is the semiconductor optical amplifier (SOA) gate switch. The diagram of a basic SOA switch is show in Fig. 2.6. Light arriving on a given input is broadcast to multiple SOAs using an optical coupler. The SOAs act as gates that can either be switched on or off. If the SOA is switched on, the incoming signal is passed to the output, otherwise the signal is blocked. The advantages of SOA switches include a fast switching time on the order of 1 ns, and the possibility of multicasting a signal to multiple outputs. A disadvantage of SOA-based switches is that the couplers re-

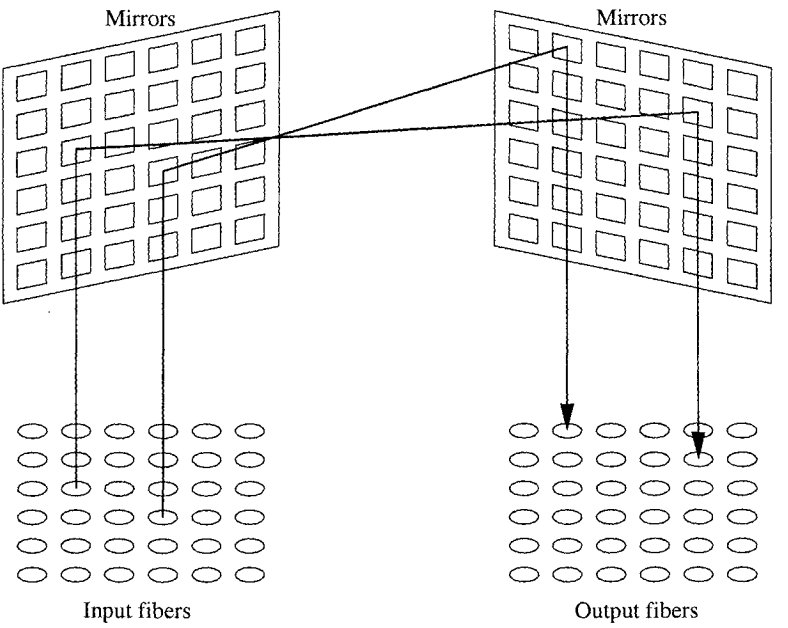


Figure 2.5. MEMS switch.

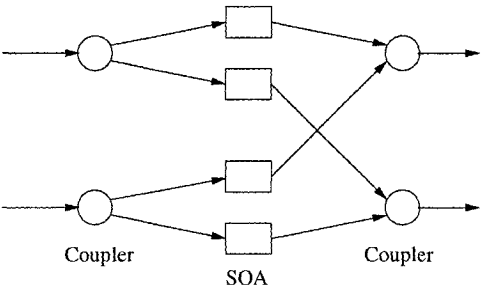


Figure 2.6. Semiconductor optical amplifier (SOA) switch.

sult in a reduction of signal power, possibly limiting the distances that signals can travel. Also, SOA devices tend to be expensive and have high polarization sensitivity [6].

2.2.2 Burst-Mode Receivers

Traditional receivers used in current optical transmission systems, such as SONET, are not well suited for optical burst switching. Such receivers assume constant phase and power for incoming signals and also assume that a signal is always present. In OBS networks, the bursts that arrive to a given receiver may have different phase and power, since

the bursts may be arriving from different sources and may be traversing different paths through the network. Furthermore, due to the nature of bursts, a signal is only present for the duration of a burst.

Burst-mode receivers are receivers that are designed to adapt to the varying phase and power of incoming bursts. Another characteristic of burst-mode receivers is their fast clock acquisition time. Burst-mode receivers that are capable of recovering the clock of an incoming 2.5 Gb/s signal within 24 ns have been demonstrated in laboratory experiments [7].

2.2.3 Wavelength Conversion

In optical burst-switched networks which utilize WDM, it is desirable to have wavelength conversion capabilities at each node in order to reduce contention. The most straightforward way to convert a signal from one wavelength to another is to covert the optical signal to an electronic signal and to use the electronic signal to modulate an optical signal on the desired output wavelength. This method is fairly simple and can convert signals that are operating at rates of up to 10 Gb/s [8]; however, the approach is not transparent and requires the optical signal to have a specific modulation format and a specific bit rate.

One approach to performing all-optical wavelength conversion is cross-gain modulation. In cross-gain modulation, the data signal is sent through a semiconductor optical amplifier (SOA) along with a continuous wave (CW) pump signal on a different wavelength. When the data signal is high, the carriers in the gain region of the SOA become depleted, and the SOA enters saturation. As a result, the amplification for the CW signal is reduced. When the data signal is low, the CW signal receives full amplification. Thus, an inverted copy of the data will be imposed on the output signal. This technique is capable of converting signals that are operating at rates of up to 10 Gb/s. The limitation of cross-gain modulation based conversion techniques is these techniques require high input power for the data signal, and the output signal has a low extinction ratio (ratio of power for a '0' bit to the power for a '1' bit). This low extinction ratio results from the fact that, even when the SOA is in saturation, the CW signal is still receiving some amount of amplification.

Another method for providing optical wavelength conversion is by using four-wave mixing. Four-wave mixing is a nonlinear effect in which signals at frequencies f_1 and f_2 interact to create new frequency components at $2f_1 - f_2$ and $2f_2 - f_1$. If the data signal is operating at frequency f_1, and if a CW pump signal is operating at frequency f_2, then the data will be imposed on new optical signals at frequencies $2f_1 - f_2$ and $2f_2 - f_1$.

The newly generated signals have lower power than the input signals; thus, the conversion efficiency for this technique is not very high. Furthermore, the efficiency decreases as the difference between the pump wavelength and the output signal wavelength increases.

2.3 Physical-Layer Issues

When designing an optical burst switched network, many physical constraints must be taken into account. Some typical physical-layer issues include attenuation, dispersion, and fiber nonlinearities. While many of these issues apply to optical networks in general, several issues may raise particular concerns in optical burst-switched networks.

2.3.1 Attenuation

As an optical signal traverses fiber, the signal power decreases due to attenuation. Attenuation is a function of the wavelength of the signal and is caused primarily by absorption and Rayleigh scattering. Absorption is caused when the light incident on silica molecules or impurities in the fiber are absorbed. For most fibers, the amount of absorption for the range of useful wavelengths (between 0.8 and 1.6 μm) is negligible. Rayleigh scattering is caused when small variations in the refractive index of the fiber scatter the light.

In an optical burst-switched network, attenuation may limit the maximum distance that a burst can travel optically. In most cases, optical amplifiers can be used to overcome attenuation; however, optical amplifiers can also introduce noise.

2.3.2 Dispersion

If an optical signal consists of multiple wavelength components, then the different components of the signal will travel at different speeds, leading to the spreading of the signal in the time domain. This effect is known as dispersion. Forms of dispersion include modal dispersion and chromatic dispersion.

Modal dispersion is caused when multiple modes of the same signal propagate at different velocities along the fiber. Modal dispersion can be eliminated by using single-mode fiber. Single-mode fiber has sufficiently small core diameter that it captures only a single fundamental mode of the propagating signal.

Chromatic dispersion is caused as a result of the speed of light in a fiber being a function of the wavelength. Thus, if the transmitted signal consists of more than one wavelength component, certain wavelength components of the signal will propagate faster than other wavelength

components, causing the signal to spread out in the time domain. Types of chromatic dispersion include material dispersion, in which the refractive index of fiber varies as a function of the wavelength, and waveguide dispersion, in which the refractive index for a particular wavelength depends on the fraction of power traveling in the core of the fiber and the fraction of power traveling in the cladding of a fiber.

For the case in which a signal consists of a pulse representing a single bit, dispersion causes the pulse to widen as it travels through a fiber. As a pulse widens, it can broaden enough to interfere with neighboring pulses (bits) on the fiber, leading to intersymbol interference. Dispersion thus limits the bit spacing and the maximum transmission rate on an optical fiber channel.

At 1300 nm, material dispersion in a conventional single-mode fiber is near zero. Fortunately, this is also a low attenuation window. Through advanced techniques such as *dispersion shifting*, fibers with zero dispersion at a specific wavelength between 1300 nm and 1700 nm can be manufactured [9]. In a dispersion-shifted fiber, the core and cladding are designed such that the waveguide dispersion is negative with respect to the material dispersion, thus canceling the total dispersion. However, the dispersion will only be zero for a single wavelength.

In addition to problems with intersymbol interference, dispersion may also introduce synchronization problems in optical burst-switched networks. In an optical burst-switched network, the burst header is typically sent on a different wavelength than the burst itself. Each of these wavelengths will experience different degrees of dispersion, causing the header and burst to either drift further apart or drift closer together in the time domain. If the physical distances of each link and the dispersion profile of the fiber are known, it may be possible to compensate for the dispersion by appropriately adjusting the offset at the source node.

2.3.3 Fiber Nonlinearities

Nonlinearities in fiber will typically have an effect on operating parameters, such as transmission rate, number of channels, channel spacing, and signal power. Examples of fiber nonlinearities include four-wave mixing, self-phase modulation, cross-phase modulation, stimulated Raman scattering, and stimulated Brillouin scattering.

Four-Wave Mixing (FWM) occurs when two wavelengths, operating at frequencies f_1 and f_2, respectively, mix to cause signals at $2f_1 - f_2$ and $2f_2 - f_1$. These extra signals, called sidebands, can cause interference if they overlap with frequencies used for data transmission. Likewise, mixing can occur between combinations of three or more wavelengths.

The effect of FWM in WDM systems can be reduced by using unequally-spaced channels [10].

Self-phase modulation is caused when changes in the intensity of a signal result in variations in the phase of a signal. The instantaneous variations in the phase of a signal can introduce additional frequency components in the signal. These additional frequency components, combined with the effects of dispersion, will lead to the spreading or compression of optical pulses in the time domain.

Cross-phase modulation is a shift in the phase of a signal caused by the change in intensity of a signal propagating at a different wavelength. Similar to self-phase modulation, the shifts in phase can introduce additional frequency components, leading to increased dispersion. Although cross-phase may limit the performance of optical communication systems, it may also have advantageous applications. Using cross-phase modulation, a signal on a given wavelength can be used to modulate a pump signal on a different wavelength. Such techniques can be used in wavelength conversion devices.

Stimulated Raman Scattering (SRS) is caused by the interaction of light with molecular vibrations. Light incident on the molecules creates scattered light at a longer wavelength than that of the incident light. A portion of the light traveling at each frequency in a Raman-active fiber is downshifted across a region of lower frequencies. The light generated at the lower frequencies is called the Stokes wave. The range of frequencies occupied by the Stokes wave is determined by the Raman gain spectrum which covers a range of around 40 THz below the frequency of the input light. In silica fiber, the Stokes wave has a maximum gain at a frequency of around 13.2 THz less than the input signal.

The fraction of power transferred to the Stokes wave grows rapidly as the power of the input signal is increased. Under very high input power, SRS will cause almost all of the power in the input signal to be transferred to the Stokes wave.

In multiwavelength systems, the shorter-wavelength channels will lose some power to each of the higher-wavelength channels within the Raman gain spectrum. To reduce the amount of loss, the power on each channel needs to be below a certain level. In [11], it is shown that in a 10-channel system with 10 nm channel spacing, the power on each channel should be kept below 3 mW to minimize the effects of SRS.

Stimulated Brillouin Scattering (SBS) is similar to SRS, except that the frequency shift is cause by acoustic interactions. In SBS, the shifted light propagates along the fiber in the opposite direction as the input signal. The intensity of the scattered light is much greater in SBS than in SRS, but the frequency range of SBS, on the order of 10 GHz, is much

lower than that of SRS. Also, the gain bandwidth of SBS is only on the order of 100 MHz.

To counter the effects of SBS, one must ensure that the input power is below a certain threshold. Also, in multiwavelength systems, SBS may induce crosstalk between channels. Crosstalk will occur when two counter-propagating channels differ in frequency by the Brillouin shift, which is around 11 GHz for wavelengths at 1550 nm. However, the narrow gain bandwidth of SBS makes SBS crosstalk fairly easy to avoid.

References

[1] Y. Xiong, M. Vanderhoute, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1838–1854, October 2000.

[2] H.M. Chaskar, S. Verma, and R. Ravikanth. A framework to support IP over WDM using optical burst switching. In *Proceedings, Optical Networks Workshop*, January 2000.

[3] S. Verma, H. Chaskar, and R. Ravikanth. Optical burst switching: a viable solution for terabit IP backbone. *IEEE Network*, 14(6):48–53, November 2000.

[4] F. Farahmand, V.M. Vokkarane, and J. P. Jue. Practical priority contention resolution for slotted optical burst switching networks. In Proceedings, First International Workshop on Optical Burst Switching (WOBS 2003), *co-located with OptiComm 2003*, October 2003.

[5] M. Dueser and P. Bayvel. Analysis of a dynamically wavelength-routed optical burst switched network architecture. *IEEE/OSA Journal of Lightwave Technology*, 20(4):574–586, April 2002.

[6] R. Ramaswami and K.N.Sivarajan. *Optical Networks: A Practical Perspective.* Morgan Kaufmann Publishers, 1998.

[7] K. V. Shrikhande, I. M. White, M. S. Rogge, F.-T. An, A. Srivasta, E. S. Hu, S.H. Yam, and L. G. Kazovksy. Performance demonstration of a fast-tunable transmitter and burst-mode packet reciever for HORNET. In *Proceedings, Optical Fiber Communication Conference (OFC)*, March 2001.

[8] S.J.B. Yoo. Wavelength conversion technologies for WDM network applications. *IEEE/OSA Journal of Lightwave Technology*, 14(6):955–966, June 1996.

[9] J. P. Powers. *An Introduction to Fiber Optic Systems.* Irwin, Homewood, IL, 1993.

[10] F. Forghieri, R. W. Tkach, A. R. Chraplyvy, and D. Marcuse. Reduction of four-wave mixing crosstalk in WDM systems using unequally spaced channels. *IEEE Photonics Technology Letters*, 6(6):754–756, 1994.

[11] A. R. Chraplyvy. Optical power limits in multi-channel wavelength-division-multiplexed systems due to stimulated Raman scattering. *Electronics Letters*, 20(2):58–59, 1984.

Chapter 3

BURST ASSEMBLY

Burst assembly is the process of assembling incoming data from the higher layer into bursts at the ingress edge node of the OBS network. As packets arrive from the higher layer, they are stored in electronic buffers according to their destination and class. The burst assembly mechanism must then place these packets into bursts based on some assembly policy.

The key parameter in burst assembly is the trigger criteria for determining when to create a burst and send the burst into the network. The trigger criterion for the creation of a burst is very important, since it controls the characteristic of the burst arrival into the OBS core. There are several types of burst assembly techniques adopted in the current OBS literature. The most common burst assembly techniques are *timer-based* and *threshold-based.*

In timer-based burst assembly approaches, a burst is created and sent into the optical network at periodic time intervals [1]. A timer-based scheme is used to provide uniform gaps between successive bursts from the same ingress node into the core networks. Here, the length of the burst varies as the load varies.

In threshold-based burst assembly approaches, a limit is placed on the maximum number of packets contained in each burst. Hence, fixed-size bursts will be generated at the network edge. A threshold-based burst assembly approach will generate bursts at non-periodic time intervals.

Both timer and threshold approaches are similar, since at a given constant arrival rate, a threshold value can be mapped to a timeout value and vice versa, resulting in bursts of similar length for each case.

The primary burst assembly parameters to be considered are the timer value, T, the minimum burst length, B_{min}, and the maximum burst length, B_{max}. B_{min} can be calculated based on the burst header pro-

cessing time at each node and the ratio of the control channels to the number of data channels in the fiber [13].

3.1 Timer and Threshold Selection

One problem in burst assembly is how to choose the appropriate timer and threshold values for creating a burst in order to minimize the packet loss probability in an OBS network. The selection of such an optimal threshold (or timer) value is an open issue. If the threshold is too low, then bursts will be short, generating increased number of bursts in the network. The higher number of bursts leads to a higher number of contentions, but the average number of packets lost per contention is less. Also, there will be increased pressure on the control plane to process the control packets of each data burst in an quick and efficient manner. If the switch reconfiguration time is non-negligible then shorter bursts will lead to lower network utilization due to the high switching time overhead for each switched (scheduled) burst. On the other hand, if the threshold is too high, then bursts will be long, which will reduce the total number of bursts injected into the network. Hence, the number of contention in the network reduces compared to the case of having shorter burst, but the average number of packets lost per contention will increase. Thus, there exists a tradeoff between the number of contentions and the average number of packets lost per contention. Hence, the performance of an OBS network can be improved if the incoming packets are assembled into bursts of optimal length. The same argument is true in a timer-based assembly mechanisms. Figure 3.1 displays the effect of varying packet arrival rate on timer-based and threshold-based aggregation techniques.

For the case in which packets have QoS restrictions, such as delay constraints, the obvious solution is to implement a timer-based scheme. In [3], a timer-based burst assembly scheme is considered for a connection-oriented wavelength-routed optical burst-switched networks. The timer values are selected based on the end-to-end delay requirements of the packets. On the other hand, if there is no delay constraint, a threshold-based scheme may be more appropriate, since having fixed-sized bursts in the network reduces the loss due to burst contentions in the network (variance in burst length is zero) [15].

Using both timeout and threshold together provides the best of both schemes, and burst generation is more flexible than having only one of the above. By calculating the optimum threshold value, calculating the minimum burst length, and using a timeout value based on the packet's delay tolerance, we can ensure that we have minimum loss while satisfying the delay requirement.

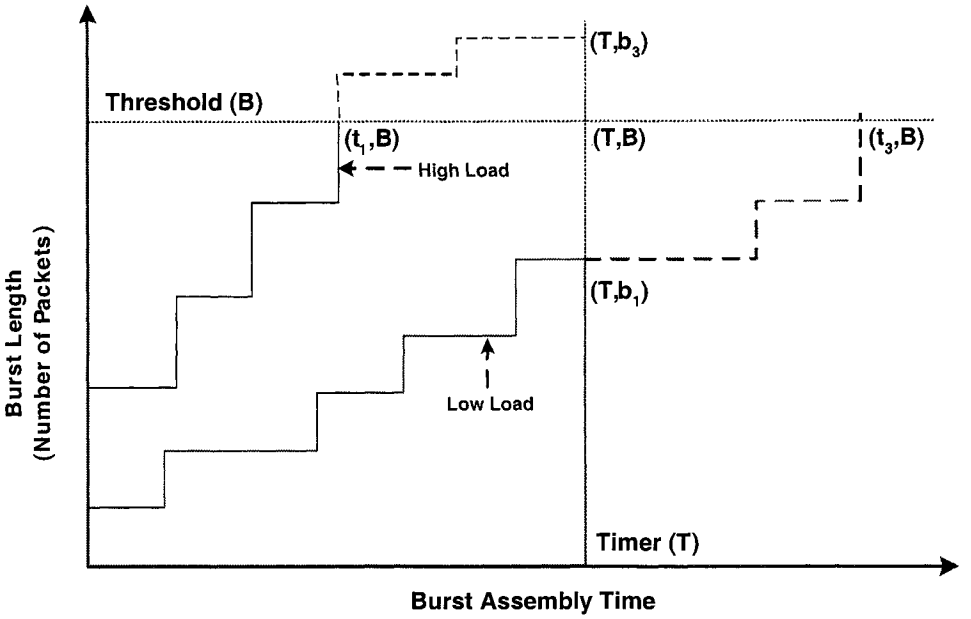


Figure 3.1. Effect of load on timer-based and threshold-based aggregation techniques.

In [5], the authors study the effect of different assembly schemes on TCP traffic. Through simulations the authors conclude that an adaptive TCP-based assembly, based on the arrival rate of TCP flows, performs better that the traditional fixed burst assembly schemes in terms of good-put and data loss rate.

The burst assembly technique adopted at the edge node has an impact on the signaling technique implemented in the core. Most signaling techniques need to know the length of the burst, the arrival time of the burst, or both in order to efficiently reserve resources in the core. For example, In JET [6], the signaling scheme needs to know both the arrival time and the length of the burst in advance. While in JIT [7, 10], no information about the burst is necessary, since the core resources are reserved in a greedy manner, leading to wastage of bandwidth at the cost of simplicity. One of the primary disadvantages of the traditional burst assembly techniques is that the signaling for resources in the core network can only be initiated after the entire burst is assembled.

In [9], a prediction-based assembly technique was proposed, in which the threshold value (or the timer value) of the next burst is predicted ahead of time based on the incoming traffic rate. Using the predicted burst length, the BHP can be sent into the core network before the actual

creation of the burst, so as to reserve the resources in the OBS core; thereby, saving on the burst assembly delay. The predicted value can be used for dynamically setting the threshold value (or timer value) for the next burst. The authors proposed a linear prediction method to predict the next burst length based on traffic correlations. The advantage of the prediction-based assembly is that the signaling and assembly can be done is parallel, thus saving on the assembly delay.

3.2 Effect of Burst Assembly on Traffic Characteristics

During burst assembly, the arriving higher-layer packets are stored in packet queues based on their destination and QoS class. After the burst creation criteria is satisfied, the corresponding burst is created and sent into the core network. Hence, we can see that the packet arrival characteristics and the packet length distribution strongly affect the corresponding burst arrival characteristics and the burst length distribution. There has been much debate as to the impact of burst assembly on the burstiness of the incoming packet traffic. It is believed that burst assembly reduces the degree of self-similarity of the input packetized traffic (smoothing effect). Note that traffic is considered to be self-similar if the arrival process is bursty at any given time scale. Traditional Poisson traffic exhibits burstiness only at smaller time scales, but approaches a constant arrival rate when considered along longer or infinite time scales. In general, it is easier to handle smoother traffic (Poisson) as compared to bursty traffic (self-similar).

The authors in [10–12] claim that burst assembly only changes the short range dependency of the input packetized traffic, but the long range characteristic on the packet traffic remains unchanged. This result contradicts the previous result presented in [13], where the authors investigate timer-based approaches for burst assembly under self-similar packet arrival patterns and show that the burst assembly mechanism reduces the self-similar characteristics of the traffic in the optical backbone.

From [10–12], the authors claim that, for a timer-based assembly scheme with a fixed burst inter-arrival distribution (T), the burst length distribution is Gaussian. Also, for a threshold-based assembly-scheme with a fixed burst length distribution (B_{max}), the burst inter-arrival distribution is Gaussian. However, the authors also mention that, although the short range dependency has a smoothing effect, timer-based and threshold-based burst aggregation techniques cannot reduce the long range dependency in a traffic process. Through simulations, the authors

of [10, 11] show that the correlation structure at large to infinite time scales still does not change.

3.3 Evaluation of Threshold-Based Burst Assembly Techniques

The work in [14] investigates threshold-based burst assembly techniques and their effect on the packet loss performance in an optical burst-switched network. The work also studies the effect of burst assembly on providing QoS support in an OBS network.

In this study, packets are assembled into bursts based on the their destination (egress router) and their QoS class, and each type of burst is assembled using a unique threshold value. Incoming packets may belong to a specific *class*, which represents the QoS requirements of the packets. Without loss of generality, we assume that there are two classes of input traffic, namely, Class 0 and Class 1, where Class 0 traffic is of higher-priority than Class 1 traffic. The objective is to find the optimal threshold range that minimizes the loss of Class 0 packets for a given network under a given load. Also, it is assumed that bursts composed of Class 0 packets are assigned a burst priority, Priority 0, and the bursts composed of Class 1 packets are assigned a burst priority, Priority 1.

An OBS network which uses the JET signaling technique with burst segmentation is considered. Bursts may receive differentiated treatment in the OBS core based on the burst priority. The network does not support fiber delay lines or wavelength converters. In the following sections, we describe the various threshold-based burst assembly techniques and show the effect of these techniques on packet loss.

3.3.1 Threshold-Based Burst Assembly Technique

For burst assembly, a threshold is used as a limiting parameter to determine when to generate a burst and send the burst into the optical core network. The threshold specifies the number of packets to be aggregated into a burst. Until the threshold condition is met, the incoming packets will be stored in prioritized packet queues at the ingress node. Once the threshold is reached, a burst is created and will be sent into the optical network. Due to the threshold policy, all bursts will have the same number of packets when entering into the network; however, as a burst traverses the OBS core, the burst length can change based on the contention resolution policies, such as burst segmentation, followed at the core.

As discussed in Section 3.1, there is a tradeoff between the number of contentions and the average number of packets lost per contention, and

it is expected that there is an optimum range of threshold values which will minimize the packet loss probability. Our primary goal is to find the optimal threshold range for a given range of load in the network.

For the case in which there are multiple classes of packets, a single threshold may be applied to all packets regardless of class, or different thresholds may be applied to each class of packets. Having multiple threshold may be essential to satisfy the QoS delay and loss guarantees of each class. In this case, the objective is to find the optimal threshold for each class of packets such that the QoS requirements are met.

In the optical core, it it possible to further differentiate between bursts that contain different classes of packets by assigning priorities to each burst and by applying prioritized contention resolution policies. By combining class-based thresholds and multiple burst priorities, we can achieve a greater degree of differentiation for different classes of traffic.

The performance of different threshold schemes are compared under the standard drop policy (DP) and the segmentation policy (SDP) for contention resolution. In the standard drop policy, the later-arriving burst is dropped if it contends with another burst. In the segmentation policy, the overlapping segments of earlier-arriving burst are dropped when the later-arriving burst contends with it.

We begin by considering one class of data traffic, and then extend the concept to two classes, showing how QoS is supported in each case. The following threshold-based QoS policies are evaluated:

- *Single threshold without burst priority:* In this policy, a single threshold is used for all the data bursts. We observe the packet loss probability and the total number of contentions are analyzed for various loads and thresholds. We expect the presence of an optimum value of threshold for a given load range and for a given network, for which the probability of packet loss will be minimum.

- *Single threshold with two burst priorities:* In this policy, we assume that the network is carrying two different classes of traffic and we have a single burst length threshold for all the traffic. We evaluate the packet loss probability and the number of contentions for variations in load and threshold. The two burst priorities are Priority 0 and Priority 1. Priority 0 represents higher-priority traffic.

- *Two threshold without burst priority:* In this policy, we assume that the network is carrying the single class of traffic. We have two different thresholds in the network, so as to evaluate the effect of different burst length thresholds on the packet loss probability and the number of contentions.

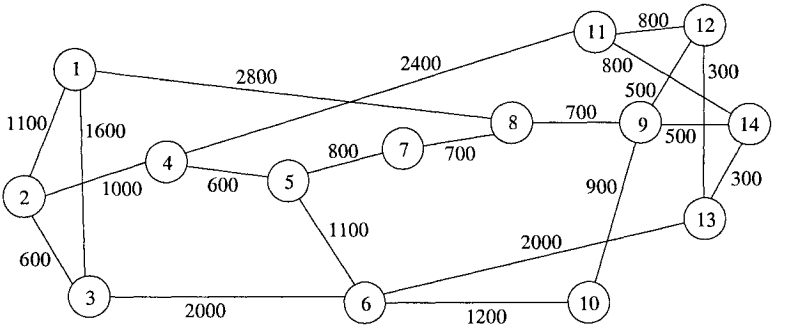


Figure 3.2. NSF network with 14 nodes (distances in km).

- *Two threshold with two burst priorities:* In this policy, we assume that the network is carrying two different classes of traffic and we have a unique burst length threshold for each class of traffic. We evaluate the packet loss probability and the number of contentions for variations in load and threshold. The two burst priorities are Priority 0 and Priority 1. Priority 0 represents higher-priority traffic.

3.3.2 Simulation Results

In order to evaluate the performance of the burst assembly technique, we develop a simulation model. The following have been assumed to obtain the results:

- Packet arrivals to the network are Poisson with rate λ.
- Packet length is fixed and is 1250 bytes.
- Transmission rate is 10 Gb/s.
- Switching time is 10 μs.
- Input traffic is uniformly distributed over all sender-receiver pairs.
- Shortest path routing is used to find the path between all node pairs.

Fig. 3.2 shows the 14-node NSF network on which the simulation was implemented. We have tested the various threshold schemes described above on the NSF network. The simulation was run until a finite number of packets were received at their destinations.

Single Threshold Without Burst Priority

In the case of a single class of packets and a single burst priority level, a single threshold is used. The packet loss probability and the total number of contentions are analyzed for various loads and thresholds. From

this single-threshold result we observe an optimum value of threshold for a given load and for a given network, for which the probability of packet loss will be minimum. Figs 3.3(a) and (b) gives the loss performance with DP or SDP as the contention resolution policy at the core.

Fig. 3.3(a) plots the total packet loss probability versus the load for threshold values of 100, 400, and 600 packets for both DP and SDP. We observe that a threshold of 400 performs better than the other two selected threshold values, 100 and 600. Hence it is essential to find an optimal threshold range to minimize loss. The need for optimal threshold can be better understood by analyzing Fig. 3.3(b). Here we observe that the loss initially decreases, hits a minimum value, and then begins to increase. The loss is minimal when the threshold value is between 380-430 packets. The initial high loss can be attributed to the loss of packets during the reconfiguration of a switch during contention resolution. The steepness in the fall of packet loss is proportional to the switching time. As the switching time becomes insignificant with respect to the burst size, the loss remains steady between the range 300-450 packets. After 450, the loss increases, since an increase in the threshold results in an increase in the average number of packets lost per contention. We choose 400 packets to be the optimal threshold value for the NSF network under a load range of 0 to 1 Erlang. The optimal threshold may vary based on the nodal degree of the network, the burst arrival rate, and the load range of the network.

Single Threshold With Burst Priority

For the case of two burst priorities and a single threshold, we evaluate the packet loss probability and the number of contentions for variations in load and threshold. The two burst priorities are Priority 0 and Priority 1. Priority 0 represents higher-priority traffic. We use the optimum threshold value obtained from Fig. 3.3(b) as the threshold value, since it minimizes packet loss. Fig. 3.4 and 3.5, give the performance with SDP as the contention resolution policy in the OBS core. We assume that the input data arrival ratio of both class of packets is the same.

Fig. 3.4 plots the packet loss probability versus load for threshold values of 100, 400, and 600 packets for both burst priorities. We observe that the packet loss for higher-class packets is significantly lower than the packet loss for lower-class packets. We observe that, even with a higher number of contentions, we achieve lower loss for higher-class packets due to segmentation.

The combined graph of packet loss probability for both Priority 0 and Priority 1 bursts is plotted versus varying threshold values in Fig. 3.5. We observe that the loss of high-class packets is lower than that of low-

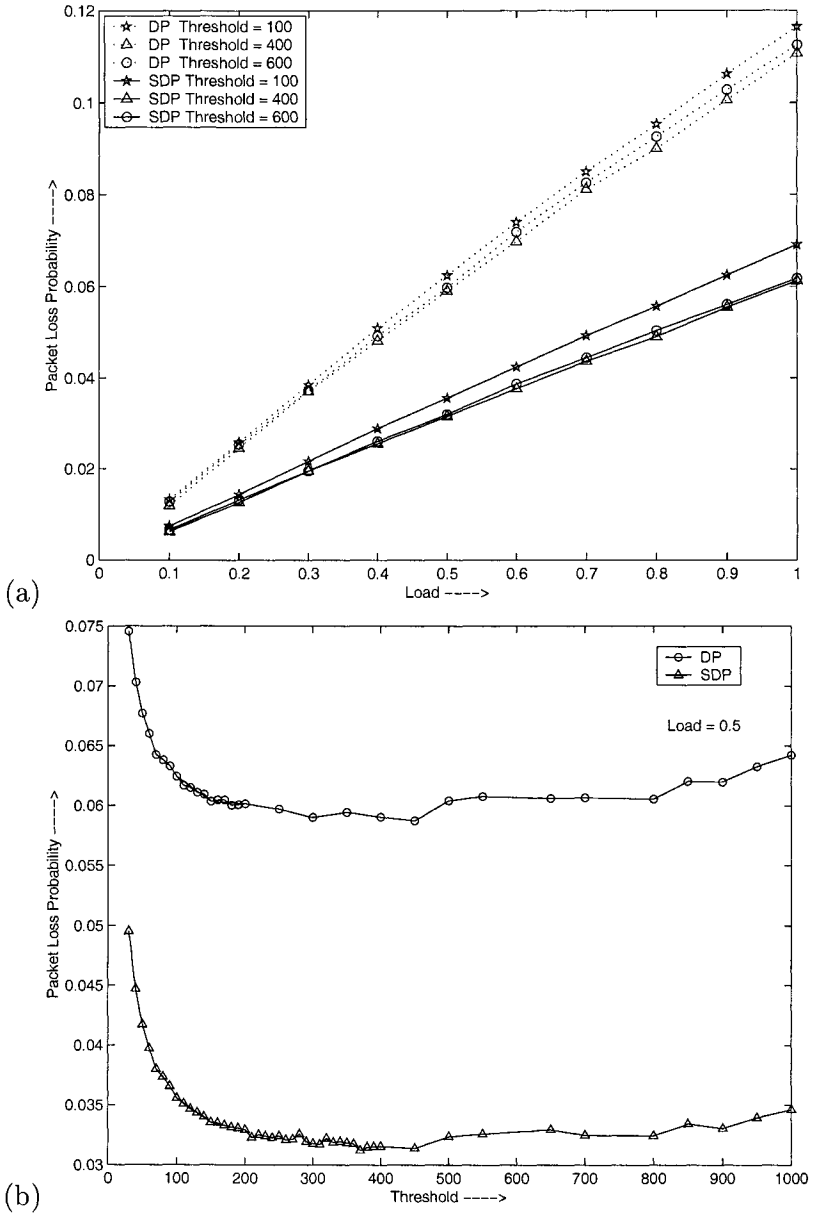


Figure 3.3. The graphs for DP and SDP with single threshold and no burst priority in the network. (a) Packet loss probability versus load. (b) Packet loss probability versus varying threshold values.

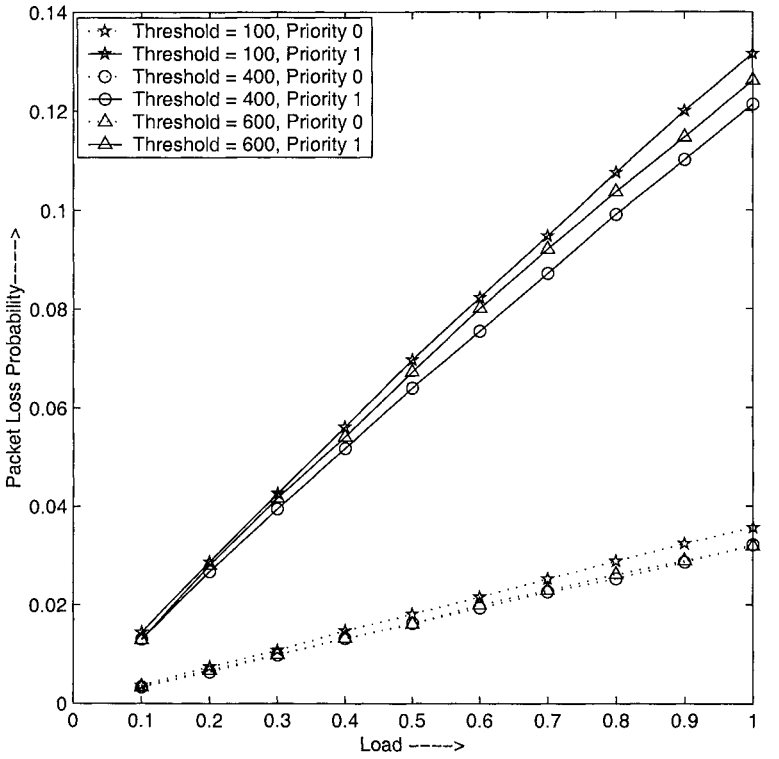


Figure 3.4. The graphs for SDP with single threshold and two burst priorities in the network. Packet loss probability versus load for different threshold values.

class packets. Also, we can see that the loss increases as the threshold value increases beyond 400 packets. We observe that Priority 0 bursts have minimum loss at threshold values of 400 and 600 packets, while Priority 1 bursts have minimal loss at a threshold of 400 packets.

In the following section, we will see that varying individual threshold values for each burst priority results in better performance for both packet classes.

Two Thresholds Without Burst Priority:

In case of two threshold values with no priorities in the bursts, we evaluate the packet loss probability and the number of contentions for variations in threshold. The results are shown in Fig. 3.6. SDP is assumed to be adopted in the core, and the network load is 0.5 Erlang. The packet arrival rate for each class of traffic is identical.

In Fig. 3.6 we observe the packet loss probability for different values of threshold. Since there are no burst priorities in the network, during

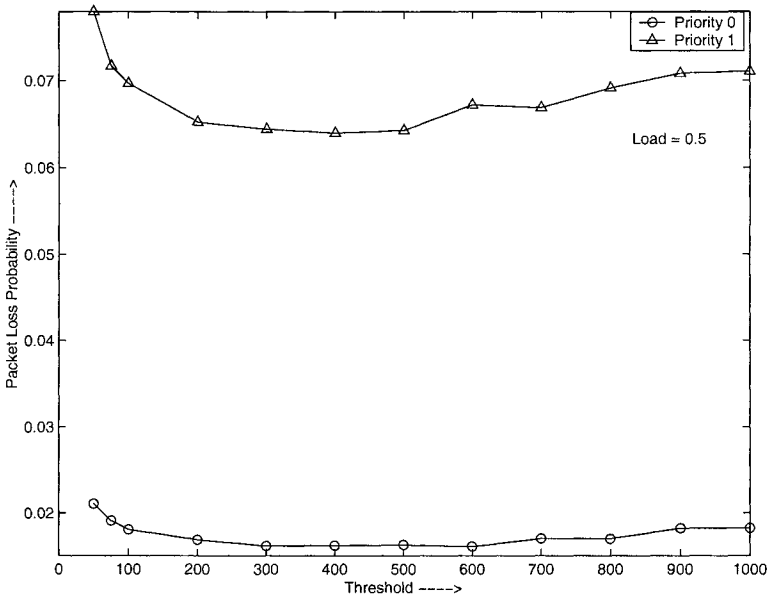


Figure 3.5. The graphs for SDP with single threshold and two burst priorities in the network. Packet loss probability versus threshold for both classes of packets at a load of 0.5 Erlang.

a contention, the burst length acts as the priority; hence longer bursts have lower loss than shorter bursts. We observe that the packet losses for the shorter burst is always higher than the packet loss for a longer burst. Therefore, the two planes in Fig. 3.6 meet when both thresholds are equal. Since no priority is incorporated into the network, the loss is symmetrical for bursts of both threshold values.

Two Thresholds With Burst Priority:

Figure 3.7 shows the network performance with two burst priorities and two threshold values, and with SDP as the contention resolution policy in the OBS core. We assume that the input data arrival ratios of both traffic classes are identical. We observe the service differentiation between the two different class of packets.

Figure 3.7 plots the packet loss probability versus varying threshold values for both priorities, under a load of 0.5 Erlang. We observe that the loss of high-class packets remains constant for different values of Threshold 1. The loss of low-class packets decreases as its burst size increases due to fewer contentions with higher-priority bursts. As the threshold increases, the loss increases due to the increase in the average number of packets lost per contention.

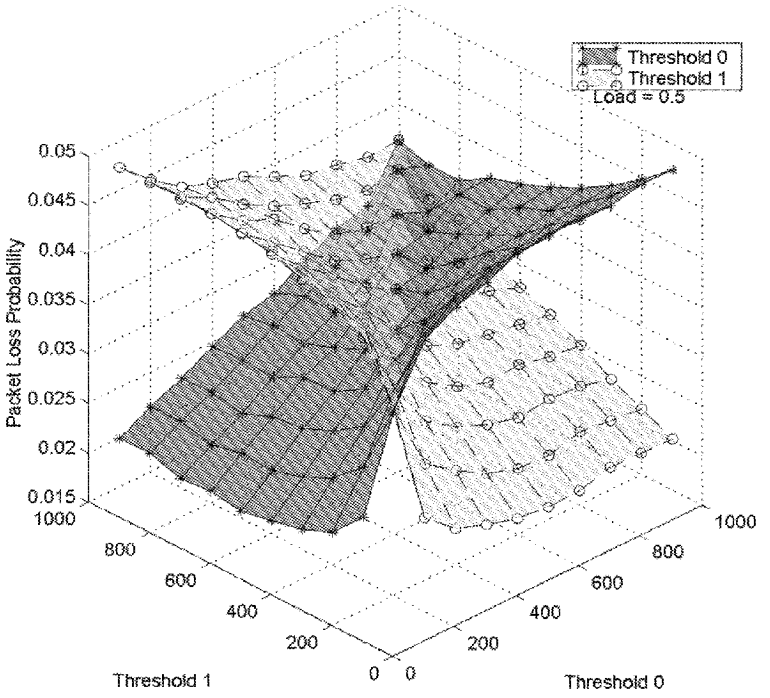


Figure 3.6. The graphs for SDP with two thresholds and no burst priority in the network Packet loss probability versus varying both threshold values for both priorities.

In general, it is observed that the average packet loss probability in the network initially decreases with the increases in burst length threshold, and reaches a minimum at the optimal threshold value. After reaching the optimum threshold value, the average packet loss probability begins to slightly increase with the increase in burst length threshold. By performing additional simulation, we have observed that when we run the simulator for 10 billion (10^{10}) fixed-size packets, the average packet loss probability remains flat after reaching an optimal threshold value. Hence, all burst which are greater than or equal to the optimal threshold value will have minimum loss. Although, by increasing the burst length threshold, we are reducing the load on the OBS control plane, we also have to consider the impact of increased burst length on end-to-end packet delay.

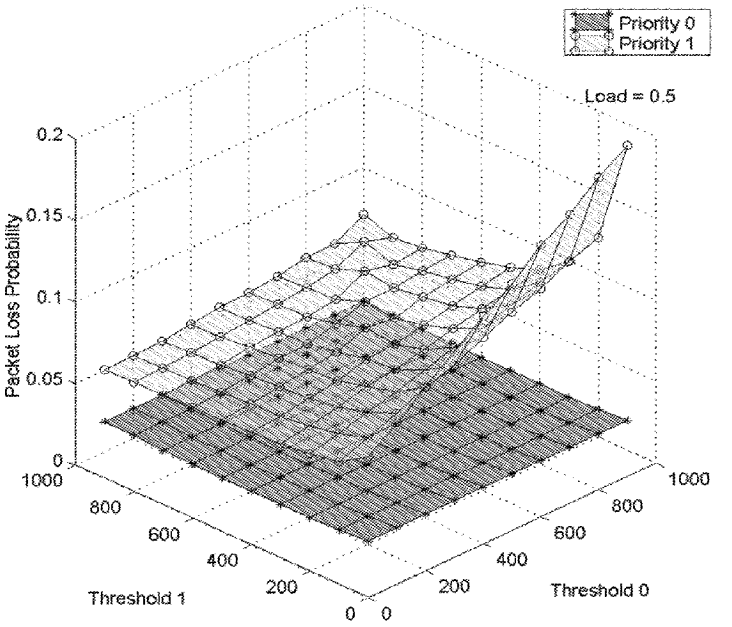


Figure 3.7. The graphs for SDP with two threshold and two burst priorities in the network Packet loss probability versus varying threshold values for both priorities.

References

[1] 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.

[2] Y. Xiong, M. Vanderhoute, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1838–1854, October 2000.

[3] M. Duser and P. Bayvel. Performance of a dynamically wavelength-routed optical burst switched network. In *Proceedings, IEEE Globecom*, volume 4, pages 2139–2143, November 2001.

[4] V. M. Vokkarane and J. P. Jue. Prioritized burst segmentation and composite burst assembly techniques for QoS support in optical burst switched networks. *IEEE Journal on Selected Areas in Communications*, 21(7):1198–1209, September 2003.

[5] X. Cao, J. Li, Y. Chen, and C. Qiao. TCP/IP packets assembly over optical burst switching network. In *Proceedings, IEEE Globecom*, volume 3, pages 2808–2812, November 2002.

[6] M. Yoo and C. Qiao. A novel switching paradigm for buffer-less WDM networks. *IEEE Communications Magazine*, 1999.

[7] J.Y. Wei, J.L. Pastor, R.S. Ramamurthy, and Y. Tsai. Just-in-time optical burst switching for multi-wavelength networks. In *Proceedings, IFIP TC6 International Conference on Broadband Communications*, pages 339–352, November 1999.

[8] 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 Magazine*, 40(2):82–89, February 2002.

[9] D. Morato, J. Aracil, L.A. Diez, M. Izal, and E. Magana. On linear prediction of Internet traffic for packet and burst switching networks. In *Proceedings, International Conference on Computer Communications and Networks (ICCCN)*, pages 138–143, 2001.

[10] X. Yu, Y. Chen, and C. Qiao. Study of traffic statistics of assembled burst traffic in optical burst switched networks. In *Proceedings, SPIE OptiComm*, pages 149–159, 2002.

[11] X. Yu, Y. Chen, and C. Qiao. Performance evaluation of optical burst switching with assembled burst traffic input. In *Proceedings, IEEE Globecom*, volume 3, pages 2318–2322, November 2002.

[12] M. Izal and J. Aracil. On the influence of self-similarity on optical burst switching traffic. In *Proceedings, IEEE Globecom*, volume 3, pages 2308–2312, November 2002.

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

[14] V. M. Vokkarane, K. Haridoss, and J. P. Jue. Threshold-based burst assembly policies for QoS support in optical burst-switched networks. In *Proceedings, SPIE OptiComm*, volume 4874, pages 125–136, July 2002.

Chapter 4

SIGNALING

When a burst is transported over the optical core, a signaling scheme must be implemented in order to allocate resources and to configure optical switches for the burst at each node. The signaling scheme in an optical burst-switched network is typically implemented using out-of-band burst header packets. In an out-of-band signaling scheme, the header associated with a burst is transmitted on a different wavelength from the burst itself. The out-of-band header packet travels along the same route as the burst, informing each node along the route to configure its optical crossconnect to accommodate the arriving burst at the appropriate time.

In this chapter, we discuss the various parameters that characterize different OBS signaling protocols. We then describe in detail several OBS signaling protocols that have been proposed in the research literature.

4.1 Classification of Signaling Schemes

Several variations of optical burst switching signaling protocols are possible, depending on how and when the resources along a route are reserved for a burst. In particular, a signaling scheme can be characterized by the following characteristics:

- one-way, two-way, or hybrid reservation;
- source-initiated, destination-initiated, or intermediate-node-initiated reservation;
- persistent or non-persistent reservation;
- immediate or delayed reservation;

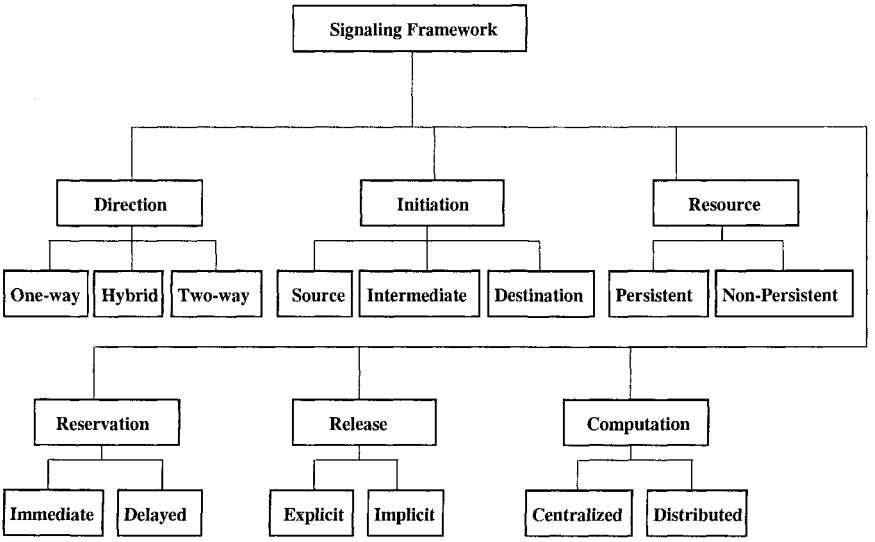


Figure 4.1. Signaling Classification.

- explicit or implicit release of resources;
- centralized or distributed signaling.

These characteristics are described in the following sections.

4.1.1 One-way, Two-way, or Hybrid

A signaling scheme can be described as operating as using either one-way reservations, hybrid reservations, or two-way reservations. In a signaling scheme with one-way reservations, the source node sends out a control packet requesting each node in the route to allocate the necessary resources for the data burst and to configure their crossconnects accordingly. The source node then sends out the data burst without waiting for an acknowledgement from either the intermediate nodes or the destination node regarding the success or failure of the resource reservation process at each node. Since the reservations are unconfirmed (one-way), it is possible that the reservations were not successful and that the burst will need to be dropped. On the other hand, by not needing to wait for an acknowledgement, the data burst can be sent out sooner, thereby reducing the end-to-end data transfer latency.

Signaling techniques with two-way reservations are acknowledgment-based. When the burst header is sent from source to destination to reserve resources for a burst, an acknowledgement message that confirms the successful assignment of requested resources is sent back from

the destination to the source. The data burst is transmitted only after the acknowledgement is received. If any of the intermediate nodes in the path are unable to accommodate the burst, then the node at which the request is blocked will send a negative acknowledgement to the source, indicating the failure of the reservation. This node will also take suitable actions to release all reservations (if any) on previous links in the path. The source can choose to retry the request by sending a new burst header, or it may simply drop the request. Signaling schemes with confirmed (two-way) reservations can eliminate the loss of bursts in the OBS core, but will also lead to higher end-to-end delay for each burst.

A hybrid signaling technique that offers a trade-off between one-way and two-way reservations is one which provides partial confirmation of reservations. In hybrid signaling schemes, the reservations from the source to some intermediate node in the route are confirmed through acknowledgements, while the reservations from the intermediate node to the destination are unconfirmed. The position of the initiating node will determine the loss and delay characteristics for a burst. If the intermediate node is closer to the source, the performance is similar to that of unconfirmed reservations, and if the intermediate node is closer to the destination, the performance is similar to confirmed reservations. This hybrid technique in described in further detail in Section 4.4.

4.1.2 Source-Initiated, Destination-Initiated, or Intermediate-Node-Initiated Reservation

A signaling technique can initiate the reservation of the requested resources at the source, at the destination, or at an intermediate hop. In the *source initiated reservation (SIR)* technique, the resources for the burst are reserved in the forward path as the burst header travels from the source to the destination. If the resource allocation is successful in the forward direction and a confirmed reservation technique is used, then an acknowledgment message indicating the reserved wavelengths is sent back to the source. The source, upon receiving the resource confirmation, transmits the burst into the core network at the scheduled time.

In a *destination initiated reservation (DIR)* technique, the source transmits a resource request to the destination node, this request collects wavelength availability information on every link along the route. Based on the collected information, the destination node will choose an available wavelength (if such exists) for the appropriate time interval, and send a reservation request back to the source node. The reservation request will traverse the intermediate nodes, reserving the chosen wavelength for the appropriate period of time. The primary cause of blocking

(or data loss) in SIR is due to the lack of free resources, while in DIR, the loss is due to outdated information [1, 2].

In an *intermediate node initiated reservation (INI)*, typically the resources are reserved similar to DIR from the source to some intermediate node, and similar to SIR from the intermediate node to the destination node.

In general, in order to reduce the loss the nodes in the forward direction, SIR based techniques may reserve more than one (or all available) wavelengths until the destination, and release the unnecessary reservation on the backward (reverse) direction. This approach may lead to lower performance due to blocking in the forward direction due to *lack of resources.* On the other hand, DIR based techniques just collect the state availability information of all intermediate nodes and then based on that information, selects a wavelength. Since, the individual state information received are not up-to-date, the selected wavelength may be taken by some other request during the time the status was collected to the time when the reservation message arrives at that node, also known as the *vulnerable period.* Hence, DIR suffers from loss due to *outdated information* during the vulnerable period.

4.1.3 Persistent or Non-persistent

One critical decision that each signaling technique needs to make is either to wait for a blocked resource (until if becomes free) or immediately indicate that there is a contention and initiate suitable connection failure mechanisms such as re-transmission, deflection, and buffering. In a persistent approach, waiting for blocked resource and assigning the wavelength results in minimum loss, assuming that suitable buffers are provisioned at the nodes (edge and core), so as to store the incoming bursts. In the non-persistent approach, the objective is to have a bound on the delay (minimize round trip delay); hence the node declares the request to be a failure if the resource is not available immediately, and implements appropriate contention resolution techniques.

4.1.4 Immediate Reservation or Delayed Reservation

Based on the duration of the reservation on the channel, the signaling techniques can be categorized as *immediate reservation* or *delayed reservation.* In the immediate reservation technique, the channel is reserved immediately from the instant that the setup message (BHP) reaches the node. On the other hand, in a delayed reservation technique, the channel is reserved from the actual arrival instant of the data burst at that

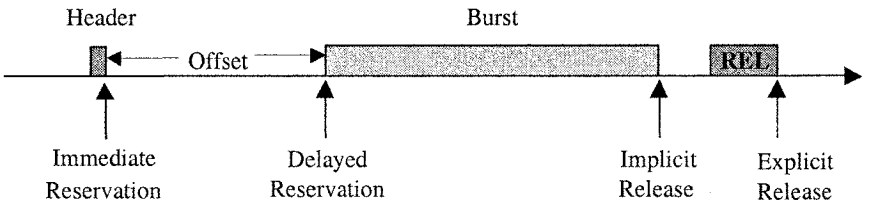


Figure 4.2. Reservation and Release Mechanisms in OBS.

node (or link). In order to employ delayed reservation, the BHP must carry the offset time between itself and its corresponding data burst. For example, the just-in-time (JIT) signaling technique uses immediate reservation, while the just-enough-time (JET) signaling technique adopts delayed reservation. In general, immediate reservation is simple and practical to implement, but incurs higher blocking due to inefficient bandwidth allocation. On the other hand, implementation of delayed reservation is more involved, but leads to higher bandwidth utilization. Delayed reservation techniques also leads to the generation of idle voids between the scheduled bursts on the data channels. Scheduling algorithm used during reservation will need to store additional information about the voids. Based on that information, the scheduler must assign a wavelength to the reservation request. Delayed reservation and immediate reservation can be incorporated into any signaling technique, if the underlying node maintains the relevant information.

4.1.5 Explicit Release or Implicit Release

An existing reservation can released in two ways, either implicitly or explicitly. In an *explicit release* technique, a separate control message is sent following the data burst, from the source towards the destination, in order to release or terminate an existing reservation. On the other hand, in an *implicit release* technique, the control message (BHP) has to carry additional information such as the burst length and the offset time. We can see that the implicit release technique results in better loss performance, due to the absence of any delay between the actual ending time of the burst and the arrival time of the release control message at each node. On the other hand, the explicit release technique results in lower bandwidth utilization and increased message complexity.

Based on the reservation and release mechanisms (Fig. 4.2), the signaling techniques can be categorized into four categories, *Immediate Reservation with Explicit Release*, *Immediate Reservation with Implicit Release*, *Delayed Reservation with Explicit Release*, and *Delayed Reser-*

vation with Implicit Release [10, 4]. Immediate reservation and explicit release indicates that an explicit control message is sent in order to perform the intended functionality, such as reserving a channel or releasing a connection. In delayed reservation, the out-of-band BHP needs to carry the offset time, and in the case of implicit release, the duration of the data burst (in addition to the offset time). We can easily observe that techniques employing delayed reservation and implicit release result in higher bandwidth utilization, while the techniques employing immediate reservation and explicit release are simple to implement at the expense of lower bandwidth utilization.

4.1.6 Centralized or Distributed

In a *centralized signaling* technique, as proposed by [5], a dedicated centralized request server is responsible for setting up the route and assigning the wavelength on each route for every data burst for all source-destination pairs in the network. The centralized technique may perform more efficiently when the network is small and the traffic is non-bursty. On the other hand, in *distributed signaling* techniques, each node has a burst scheduler that assigns an outgoing channel for each arriving BHP in a distributed manner. The distributed approach is suitable of large optical networks and for bursty data traffic.

The objective of having a generalized signaling framework is that we can now categorize each signaling technique based on the parameter selections made, and the corresponding performance of the technique can be deduced.

Two prominent signaling techniques for a bufferless OBS network are Tell-and-Wait (TAW) and Just-Enough-Time (JET). In both of these techniques, a BHP is sent ahead of the data burst in order to configure the switches along the burst's route. We now describe these two signaling techniques.

4.2 Just-Enough-Time (JET)

Figure 4.3 illustrates the JET signaling technique. As shown, a source node first sends a burst header packet (BHP) on a control channel toward the destination node. The BHP is processed at each subsequent node in order to establish an all-optical data path for the corresponding data burst. If the reservation is successful, the switch will be configured prior to the burst's arrival. Meanwhile, the burst waits at the source in the electronic domain. After a predetermined offset time, the burst is sent optically on the chosen wavelength [6]. The offset time is calculated based on the number of hops from source to destination, and the switch-

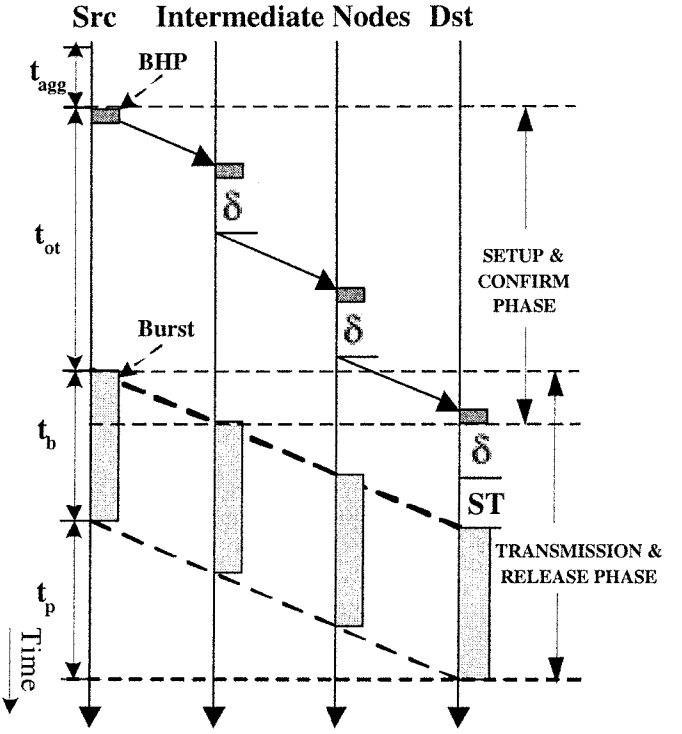


Figure 4.3. Just-Enough-Time (JET) signaling technique.

ing time of a core node. Offset time is calculated as $OT = h \cdot \delta + ST$, where h is the number of hops between the source and the destination, δ is the per-hop burst header processing time, and ST is the switching reconfiguration time. If at any intermediate node, the reservation is unsuccessful, the burst will be dropped. The unique feature of JET when compared to other one-way signaling mechanisms is delayed reservation and implicit release.

The information necessary to be maintained for each channel of each output port of every switch for JET comprises of the starting and the finishing times of all scheduled bursts, which makes the system rather complex. On the other hand, JET is able to detect situations where no transmission conflict occurs, although the start time of a new burst may be earlier than the finishing time of an already accepted burst, i.e., a burst can be transmitted in between two already reserved bursts. Hence, bursts can be accepted with a higher probability in JET.

There are other closely related one-way based signaling techniques, such as Tell-and-Go (TAG) and Just-in-Time (JIT). In the TAG ap-

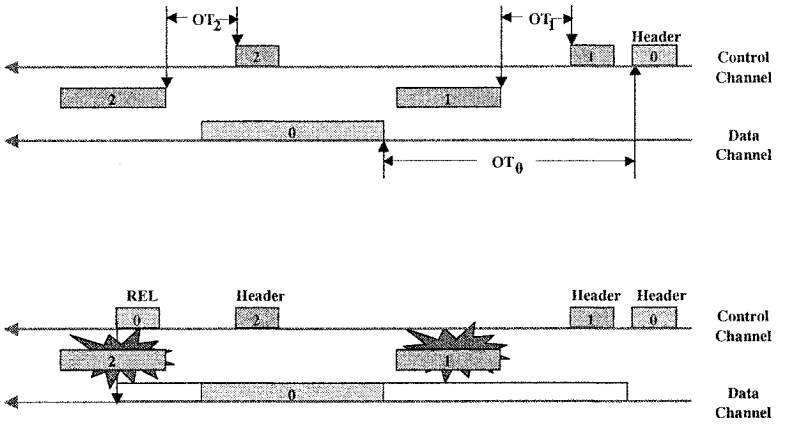


Figure 4.4. Comparison of (a) JET and (b) JIT based signaling.

proach, the data burst must be delayed at each node in order to allow time for the burst header to be processed and for the switch to be configured, instead of pre-determining this duration at the source and incorporating the delay in the offset time. This delay requires the use of *fiber delay lines* (FDL), which consist of loops of optical fiber. The propagation delay in the FDL is the amount of time for which the data burst will be delayed.

JIT is similar to JET except that JIT employs immediate reservation and explicit release instead of delayed reservation and implicit release. Fig. 4.4(a) and (b) compares a similar signaling scenario using JET and JIT, respectively. An architectural framework for implementing various JIT schemes is presented in [10]. The primary benefit of using these one-way based techniques is the minimized end-to-end delay for data transmission over an optical backbone network, at the cost of high packet loss due to data burst contentions for resources at the bufferless core network.

4.3 Tell-and-Wait (TAW)

Figure 4.5 illustrates the TAW signaling technique. In TAW, the "SETUP" BHP is sent along the burst's route to collect channel availability information at every node along the path. At the destination, a channel assignment algorithm is executed, and the reservation period on each link is determined based on the earliest available channel times of all the intermediate nodes. A "CONFIRM" BHP is sent in the reverse direction (from destination to source), which reserves the channel for the requested duration at each intermediate node. At any node along the

path, if the required channel is already occupied, a "RELEASE" BHP is sent to the destination to release the previously reserved resources. If the "CONFIRM" packet reaches the source successfully, then the burst is sent into the core network.

Also, since TAW is similar to wavelength-routed networks, the channel can be reserved in the forward direction as in source initiated reservation (SIR) or in the reverse direction from the destination back to the source as in *destination initiated reservation (DIR)* [2, 1]. TAW in OBS is different from wavelength-routed WDM networks in the sense that in TAW, resources are reserved at any node only for the duration of the burst. Also, if the duration of the burst is known during reservation, then an implicit release scheme can be followed to maximize bandwidth utilization.

All the protocols discussed above are one-way signaling techniques except TAW, which is a two-way signaling technique. If we compare TAW and JET, the disadvantage of TAW is the round-trip setup time, i.e., the time taken to set up the channel; however in TAW the data loss is very low. Therefore TAW is good for loss-sensitive traffic. On the other hand, in JET, the data loss is high, but the end-to-end delay is less than TAW. In TAW, it takes three times the one-way propagation delay from source to destination for the burst to reach destination, whereas in the case of JET, the delay is just the sum of one one-way propagation delay and an offset time. There is no signaling technique that offers the flexibility in both delay and loss tolerance values.

4.4 Intermediate Node Initiated (INI) Signaling

Several signaling techniques have been proposed for transmitting data all-optically in an OBS networks. To accommodate the dynamic resource reservation requests to transmit data bursts, the signaling technique has to first find a route from the source to the destination, then schedule the burst on a particular wavelength at each intermediate node.

The most commonly studied distributed signaling techniques are tell-and-wait (TAW) and just-enough-time (JET). TAW is a two-way, acknowledgment based signaling technique using explicit setup and release control messages. JET is a one-way based signaling technique without acknowledgments that uses estimated setup and release burst header packets (BHPs). In order to avoid converting to electronics in the core, all signaling techniques have an offset time between the BHP and the corresponding data. The BHP may also specify the duration of the burst in order to let a node know when it may reconfigure its switch for the next burst [7], in addition to containing the offset time. The offset time

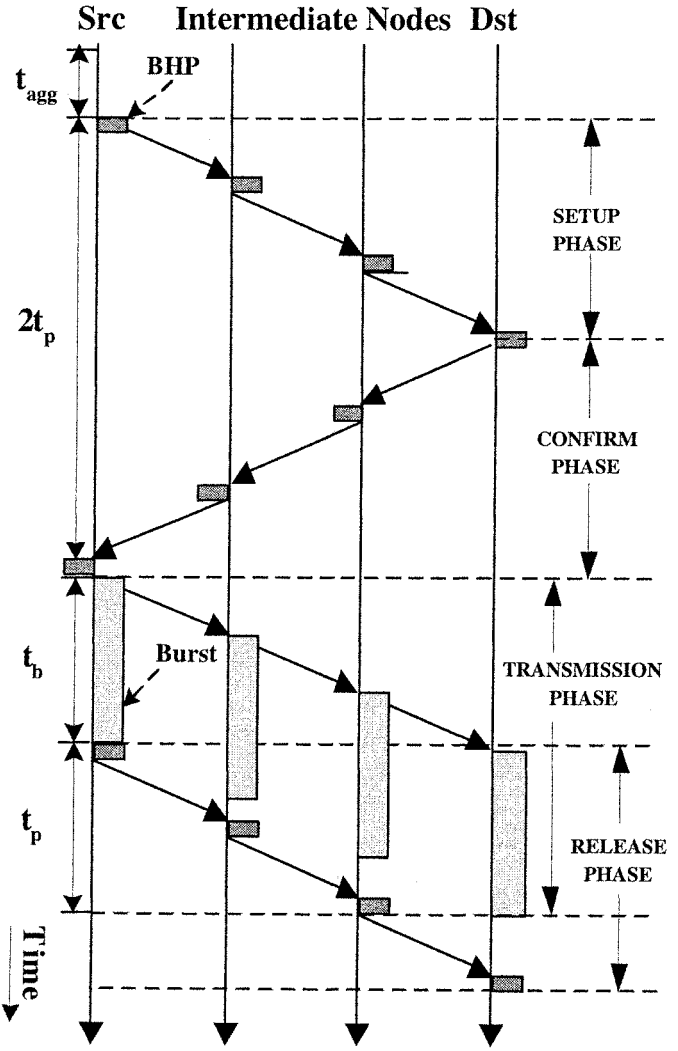


Figure 4.5. Tell-and-Wait (TAW) signaling technique.

allows for the BHP to be processed at each intermediate node before the burst arrives at the intermediate node.

If we compare TAW and JET, the disadvantage of TAW is the round-trip setup time, i.e., the time taken to set up the channel; however in TAW the data loss is very low. Therefore TAW is good for loss-sensitive traffic. On the other hand, in JET, the data loss is high, but the end-to-end delay is less than TAW. In TAW, it takes three times the one-way propagation delay from source to destination for the burst

to reach destination, whereas in the case of JET, the delay is just the sum of one one-way propagation delay and an offset time. There is no signaling technique that offers the flexibility in both delay and loss tolerance values.

In an IP over OBS network, it is desirable to provide QoS support for applications with diverse QoS demands, such as voice-over-IP, video-on-demand, and video conferencing. Several solutions have been proposed to support QoS in the OBS core network (refer to Chapter 7). There is no single technique that offers flexibility to support both delay-sensitive and loss-sensitive traffic in the same OBS network. Also the existing schemes for QoS, such as JET with additional-offset time for different classes of traffic, suffer from high blocking probability. Also, the source node must estimate the offset times in order to support different packet class requirements.

In this section, we discuss a hybrid signaling technique called *intermediate node initiated (INI)* signaling, and an extension of the INI signaling technique in order to provide differentiated signaling based on application requirements through the *differentiated INI (DINI)* technique. The DINI technique provides differentiation without introducing any additional offset time.

4.4.1 Intermediate Node Initiated (INI) Signaling

In [8], in order to overcome the limitations of TAW and JET, the authors propose the intermediate node initiated signaling technique. In the INI signaling technique, a node between source and destination on the path is selected as the initiating node. An initiating node is an intermediate node between the source and the destination at which a channel reservation algorithm is run to determine the earliest time that the burst can be sent from the source node and the corresponding earliest times at which the nodes between source and the initiating node can be scheduled to receive the burst. At the initiating node, the actual reservation of the channels starts in both directions i.e., from the initiating node to the source and from the initiating node to the destination. The selection of the initiating node is critical in the INI signaling technique.

Figure 4.6 illustrates the INI signaling technique. When a burst is created at the edge node, a "SETUP" BHP is sent to the destination. The BHP collects the details of channels at every node along the path until it reaches the initiating node. At the initiating node, a channel assignment algorithm is executed to determine the time duration that the channels will need to be reserved at each intermediate hop between the source and initiating node. A "CONFIRM" packet is then sent to the source node, which reserves channels along the path from the initiating

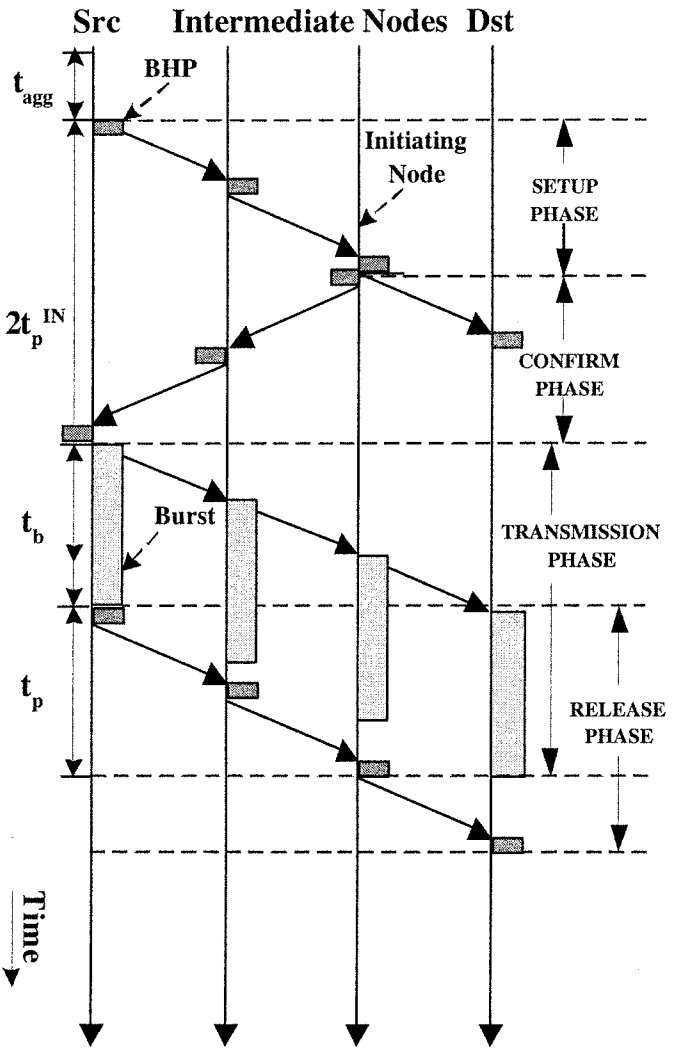


Figure 4.6. Intermediate Node Initiated (INI) Signaling Technique.

node to the source. If a channel is busy at any node, a "RELEASE" packet is sent back to the initiating node to release previously reserved resources. If the "CONFIRM" packet reaches the source successfully, then the burst is sent at the scheduled time. The IN simultaneously sends an unacknowledged "SETUP" BHP towards the destination, for reserving the channels between the IN and the destination. If, at any node between the initiating node and the destination node, the BHP fails to reserve the channel, the burst is dropped at that node.

Table 4.1. Summary of the different OBS signaling techniques.

Signaling	Direction	Initiation	Reservation	Release	Delay	Loss
TAW	two-way	src./dest.	explicit	explicit	high	low
TAG	one-way	source	implicit	implicit	least	high
JET	one-way	source	implicit	implicit	low	high
JIT	one-way	source	explicit	explicit	low	high
INI	hybrid	intermed.	exp./imp.	exp./imp.	flexible	flexible

In TAW, there is an acknowledgment from the destination before the burst is sent from the source, and in JET, there is no acknowledgment. In INI, there is an acknowledgment from the initiating node, thereby decreasing the probability of blocking compared to JET. Also since the burst waits at the source for a time less than the propagation delay from the source to the destination, INI decreases the end-to-end delay compared to TAW. In the INI signaling technique, if the initiating node is the source, then the signaling technique is identical to JET, and if the initiating node is the destination, then the signaling technique is identical to TAW. For the INI signaling technique, TAW and JET and the two extremes, so by appropriately selecting the initiating node, we can implement TAW and JET by using INI. In INI, we can use both regular reservation and delayed reservation. With delayed reservation the performance of the signaling technique improves. In the following simulations, we adopt the delayed reservation technique.

Table 1 gives the summary of the three signaling techniques in terms of burst loss probability and average end-to-end delay.

Illustration: Consider the path 2-4-5-7 in Fig. 4.7, with Node 2 as the source and Node 7 as the destination. Here we have four possible initiating nodes including the source and destination nodes. If we choose the source i.e., Node 2 as the initiating node, then the INI signaling technique becomes JET. If we choose the destination i.e., Node 7 as the initiating node, then the INI signaling technique becomes TAW. Other possibilities of initiating nodes are Node 4 and Node 5. Let us consider Node 5 to be the initiating node and observe how the INI signaling technique works. Node 2 sends the BHP to the next hop, Node 4, along with the channel availability information of the Link 2-4. Node 4 adds the channel availability information of Link 4-5 and sends the BHP to the next node, Node 5. When the initiating Node 5 gets the BHP, it runs a channel reservation algorithm to determine the earliest times at which the required burst can be served by the intermediate

nodes between the source and the initiating node, including both the source and the initiating node. A reply packet, which reserves the channels at the intermediate nodes at the pre-determined times is sent from initiating node to the source. As soon as the reply packet reaches the Source 2, the burst is sent. The BHP sent from the initiating Node 5 to the destination reaches Node 7 and configures Node 7 to receive the incoming burst at the corresponding time. Node 5 will not send any acknowledgment back to the initiating node. The BHP sent from the initiating node just reserves the available channels and proceeds in the forward direction from the initiating node to the destination.

4.4.2 Differentiated Intermediate Node Initiated (DINI) Signaling

The INI signaling technique can be extended to provide QoS at the optical layer. It is possible to implement multiple signaling techniques in the same network to provide differentiated services, in order to support both loss and delay sensitive traffic, i.e., we can use TAW for loss sensitive traffic, and JET for delay sensitive traffic. This approach of having a hybrid core network with two different signaling schemes can only provide a coarse QoS guaranty. In order to provide a finer level of QoS differentiation, we modify the INI scheme.

Using INI, we can satisfy both the loss and delay constraints of each specific application by carefully selecting the initiating node. In general, for applications with delay constraints we choose the initiating node to be closer to the source node, such that the end-to-end delay is less than the application-specified constraint. For applications with loss constraints, we choose the initiating node to be closer to the destination node, such that most of the path is two-way acknowledged.

Suppose, we have to support three classes of traffic, say P0, P1, and P2, with P0 being delay sensitive, P1 being both delay and loss sensitive, and P2 being loss sensitive. We can use the source node as the initiating node for P0, the center node as the initiating node for P1, and the destination node as the initiating node for P2, thus providing differentiated services in the same OBS network.

4.5 Analytical Delay Model

In this section, we discuss an analytical model for evaluating the delay characteristics of each OBS signaling techniques. We assume that fixed shortest-path routes, R_{sd}, are calculated by each source-destination pair; no optical buffering (FDLs) or wavelength conversion is supported at core nodes. Without loss of generality, we investigate a network with

a single wavelength per fiber. The model can be directly extended to multiple wavelengths per fiber. Due to the absence of wavelength converters, multiple wavelengths in each fiber can be thought of as multiple layers of the network, with one layer for each wavelength. It is important to compare the end-to-end delay of each signaling technique, such as JET, TAW, and INI. We begin by defining the following notation:

t_{bhp}: burst header packet (BHP) processing delay at each OBS node. We assume that the processing delays of different signaling messages, such as "SETUP", "RELEASE", and "CONFIRM", at all the nodes are identical.

t_{sw}: switching time required to reconfigure the optical cross-connect at each OBS node.

t_{agg}: burst aggregation delay based on the assembly technique adopted at the ingress OBS node.

t_b: data burst transmission time.

t_{ot}: offset time, the fixed initial time between the out-of-band BHP and the data burst at the ingress node.

t_p^{ij}: propagation delay on the fiber between Node i and Node j.

The typical values of t_p^{ij} is 5 μs/km, t_{bhp} is hundreds ns, and t_{sw} is few μs.

We first calculate the average end-to-end packet delay, T_{SIG}, incurred by each signaling technique. T_{SIG} is the duration from the instant the first packet arrives at the ingress node to the instant the burst is completely received at the destination. Consider a route with n hops to the destination.

(a) Just-Enough-Time (JET) or Just-In-Time (JIT):

In Just-Enough-Time (JET) or Just-In-Time (JIT), the end-to-end delay is given by the sum of the burst aggregation time, the offset time, the burst transmission time, and the data burst propagation time.

$$T_{JET} = T_{JIT} = t_{agg} + t_{ot} + t_b + \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} \tag{4.1}$$

where,

$$t_{ot} = n t_{bhp} + t_{sw}. \tag{4.2}$$

If we consider Tell-and-Go (TAG) signaling technique, there is a slight variation in the delay parameters, the offset time, $t_{ot} = 0$, and there is

an additional compensating per-hop FDL delay, t_{fdl} equivalent to the $t_{bhp} + t_{sw}$, that is provided by input FDLs to all the data channels at each node, so as to compensate for the control header processing and switching delay.

$$T_{TAG} = t_{agg} + nt_{fdl} + t_b + \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij}. \tag{4.3}$$

(b) Tell-and-Wait (TAW):

In Tell-and-Wait (TAW), the end-to-end delay is given by the sum of the burst aggregation time, the round trip connection setup time, the burst transmission time, and the data burst propagation time. Additional offset time may be required, if the sum of the per-hop BHP processing times at all the intermediate nodes plus one switch reconfiguration time is greater than the round-trip connection setup time. Therefore,

$$T_{TAW} = t_{agg} + 3 \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} + t_b + t_{ot}. \tag{4.4}$$

Also,

$$t_{ot} = 0 \; if \; \; 2 \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} \geq (n+2)t_{bhp} + t_{sw}. \tag{4.5}$$

(c) Intermediate Node Initiated (INI):

In INI, the end-to-end delay is given using a combination of the delay equation of TAW and JET. The end-to-end delay in INI also depends upon the location of the initiation node (IN), k, the burst aggregation time, the burst transmission time, and the data burst propagation time. Let l is the number of hops between the source and IN, and m is the number of hops between IN and destination node.

$$T_{INI} = t_{agg} + 2 \sum_{l^{ij} \in R_{sk}}^{n} t_p^{ij} + \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} + t_b + t_{ot} \tag{4.6}$$

where,

$$t_{ot} = 0 \; if \; \; 2sum_{l^{ij} \in R_{ks}}^{n} t_p^{ij} \geq \; mt_{bhp} + t_{sw} \tag{4.7}$$

else,

$$t_{ot} = (mt_{bhp} + t_{sw}) - 2 \sum_{l^{ij} \in R_{ks}}^{n} t_p^{ij}. \tag{4.8}$$

If $l = n$, then delay is same as TAW, and if $l = 0$ or $m = n$, then delay same as JET (or JIT).

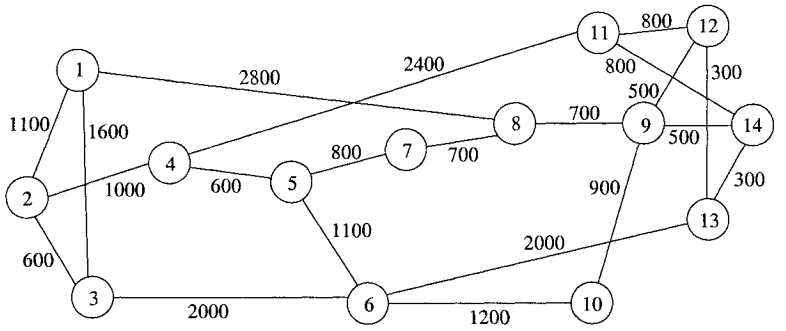


Figure 4.7. 14-node NSF backbone network topology (distance in km).

Hence,

$$T_{JET} \leq T_{INI} \leq T_{TAW}. \tag{4.9}$$

4.6 Numerical Results

In order to evaluate the performance of the INI signaling technique, a simulation model is developed. Burst arrivals to the network are Poisson, with exponentially distributed burst length, with average burst length of 0.1 ms. The link transmission rate is 10 Gb/s. Each packet is of length 1250 bytes. The switching reconfiguration time is 0.01 ms. There is no buffering or wavelength conversion at nodes. Retransmission of the lost bursts is not considered. Fig. 4.7 shows the 14-node NSFNET on which the simulation is implemented.

Figures 4.8(a) and 4.8(b) plot the burst loss probability and average end-to-end delay versus load when the initiating nodes are taken as source (SRC), first-hop (Hop-1), second-hop (Hop-2), third-hop (Hop-3), and destination (DST) respectively. In Figs. 4.8 (a) and 4.8 (b), only paths that are more than or equal to three hop counts are considered to show the effect of INI signaling technique. We observe that the loss probability decreases as the initiating node moves away from the source. If the initiating node is chosen closer to the source, a greater part of the path is unacknowledged, which leads to a higher loss probability. On the other hand, if the initiating node is chosen closer to the destination, a greater part of the path is acknowledged, which leads to a lower loss probability. We also observe that the delay increases proportionally to the increase in distance between the initiating node and the source, since the path from source to the initiating node is acknowledged, and hence incurs a higher round-trip delay. Also, the values of loss and delay when the initiating node is at the source and the destination are consistent with JET and TAW respectively.

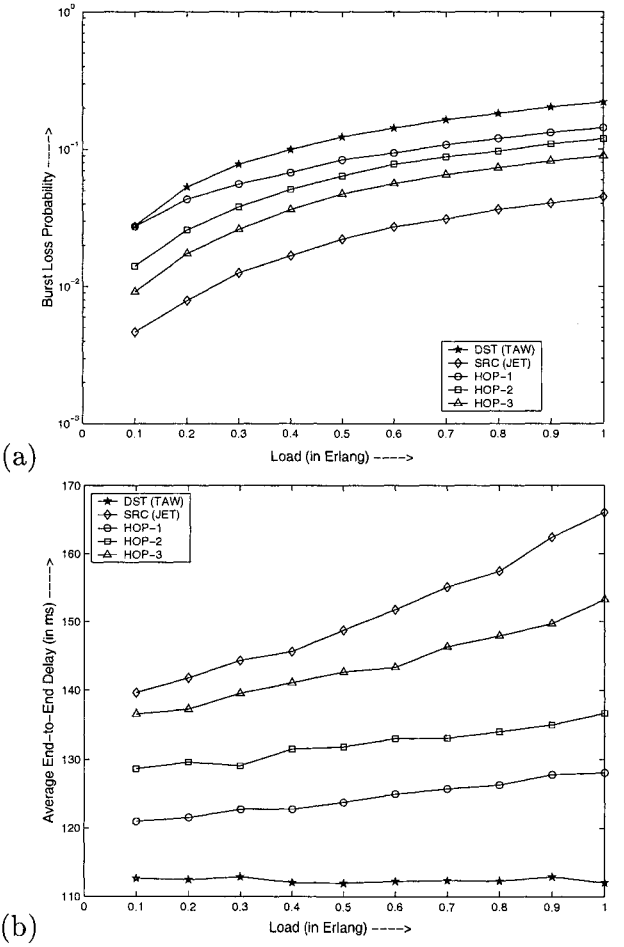


Figure 4.8. (a) Burst loss probability versus load, and (b) Average end-to-end delay versus load, when the initiating nodes are source, first hop, second hop, third hop, and destination.

Figures 4.9(a) and 4.9(b) plot the burst loss probability and average end-to-end delay versus load for the three priority bursts. We observe that P2 suffers the least loss, while P0 incurs the least delay, and P1 experiences loss and delay between the values of P0 and P2. For comparable values of offset time, we found that INI out-performs the traditional offset-based QoS scheme [7]. In the offset-based scheme, the source has to estimate the additional-offset to provide differentiated services, while in INI, the initiating node has the channel availability information of all nodes between itself and the source. Also, the data burst does not enter

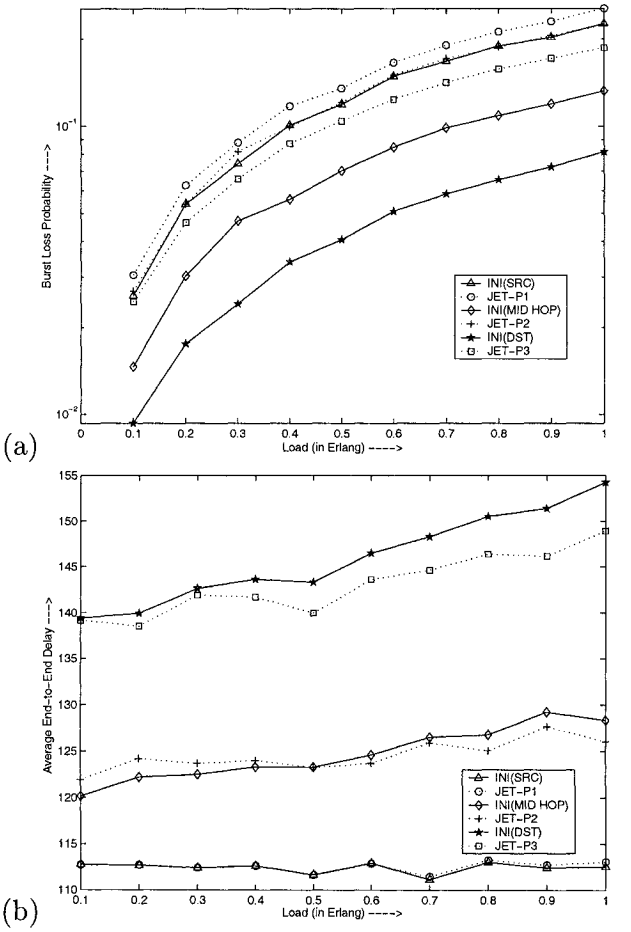


Figure 4.9. (a) Burst loss probability versus load, and (b) Average end-to-end delay versus load, when the initiating nodes is source, center hop, and destination in the same network to provide differentiation through signaling.

the network until resources have been reserved between the source node and the initiating node.

The INI signaling technique provides flexibility during channel reservation based on the type of data to be transmitted. The packet loss probability of INI is less than that of JET and the end-to-end delay is less than that of TAW. Hence, the hybrid INI technique is a flexible solution suitable for handling the varying traffic demands of the next-generation optical network.

References

[1] K. Lu, G. Xiao, and I. Chlamtac. Analysis of blocking probability for distributed lightpath establishment in WDM optical networks. *IEEE/ACM Transactions on Networking*, 2004.

[2] X. Yuan, R. Melhem, R. Gupta, Y. Mei, and C. Qiao. Distributed control protocols for wavelength reservation and their performance evaluation. *Photonic Networks and Communications*, 1(3):207–218, 1999.

[3] 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 Magazine*, 40(2):82–89, February 2002.

[4] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, and D. Stevenson. The JumpStart just-in-time signaling protocol: A formal description using EFSM. *Optical Engineering*, 42(2):568–585, February 2003.

[5] M. Dueser and P. Bayvel. Analysis of a dynamically wavelength-routed optical burst switched network architecture. *IEEE/OSA Journal of Lightwave Technology*, 20(4):574–586, April 2002.

[6] 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.

[7] C. Qiao and M. Yoo. Choices, features and issues in optical burst switching. *SPIE Optical Networks Magazine*, 1(2):36–44, 2000.

[8] R. Karanam, V. M. Vokkarane, and J. P. Jue. Intermediate node initiated (INI) signaling: A hybrid reservation technique for optical burst-switched networks. In *Proceedings, Optical Fiber Communication Conference (OFC)*, March 2003.

[9] M. Yoo, C. Qiao, and S. Dixit. QoS performance of optical burst switching in IP-over-WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2062–2071, October 2000.

Chapter 5

CONTENTION RESOLUTION

Since optical burst-switched networks provide connectionless transport, there exists the possibility that bursts may contend with one another at intermediate nodes. Contention will occur if multiple bursts from different input ports are destined for the same output port at the same time. Typically, contention in traditional electronic packet-switching networks is handled through buffering; however, in the optical domain, it is more difficult to implement buffers, since there is no optical equivalent of random-access memory. In this chapter, we discuss several possible methods for resolving contention in OBS networks.

5.1 Optical Buffering

Contention in traditional electronic packet-switching networks is implemented by storing packets in random-access memory (RAM) buffers; however, RAM-like buffering is not yet available in the optical domain. In optical networks, *fiber delay lines (FDLs)* [1–5] can be utilized to delay packets for a fixed amount of time. By implementing multiple delay lines in stages [3] or in parallel [4], a buffer may be created that can hold a burst for a variable amount of time. Some papers have investigated approaches for designing larger buffers without a large number of delay lines [6, 7]. In [6], the buffer size is increased by cascading multiple stages of delay lines. In [7], the buffer size is increased by utilizing so called non-degenerate buffers in which the length of the delay lines may be greater than the number of delay lines in the buffer. This approach yields lower data loss probabilities, but does not guarantee the correct ordering of the packets. Note that, in any optical buffer architecture, the size of the buffers is severely limited, not only by signal quality concerns, but also by physical space limitations. To delay a single burst

for 1 ms requires over 200 km of fiber. Due to the size limitation of optical buffers, a node may be unable to effectively handle high load or bursty traffic conditions. Wavelength controlled fiber loop buffers and wavelength routing based photonic packet buffers are described in [8, 9].

Optical buffers are either single-stage, which have only one block of delay lines, or multistage which have several blocks of delay lines cascaded together, where each block contains a set of parallel delay lines. Optical buffers can be further classified into feed-forward, feedback, and hybrid architectures [1, 10]. In a feed-forward architecture, each delay line connects an output port of a switching element at a given stage to an input port of another switching element in the next stage. In a feedback architecture, each delay line connects an output port of a switching element at a given stage to an input port of a switching element in the same stage or a previous stage. In a hybrid architecture, feed-forward and feedback buffers are combined. According to the position of the buffers, packet switches are essentially categorized into three major configurations: input buffering, output buffering, and shared buffering. In input buffering, a set of buffers is dedicated for each input port. In output buffering, a set of buffers is dedicated for each output port. In shared buffering, a set of buffers can be shared by all switch ports. Input buffering has poor performance due to head-of-line (HOL) blocking. Output buffering and shared buffering can both achieve good performance in any packet switch. However, output buffering requires a significant number of FDLs as well as larger switch sizes. With shared buffering, on the other hand, all output ports can access the same buffers. Therefore, it can be used to reduce the total number of buffers in a switch while achieving a desired level of packet loss. In the optical domain, shared buffering can be implemented with one-stage feedback recirculation buffering [1, 11, 12] or multistage feed-forward shared buffering [2, 3, 5]. Furthermore, buffers can be either configured as *degenerate buffer* (linear increment) or *non-degenerate buffer* (non-linear increment) [13, 7].

In addition to buffering bursts optically, it is also possible to buffer bursts electronically. Electronic buffering can be accomplished by sending the bursts up to the electronic switching or routing layer. The disadvantage of such an approach is that the network loses transparency, and each node must have electronic switching or routing capabilities, resulting in higher network costs and also requiring electronic memories which must keep up with the speeds of optical networks. Furthermore, a greater load will be placed on the processing capabilities of the electronic switch or router. An alternative would be to implement electronic buffers directly as a part of the optical switch itself. In this case each node would still require additional transmitters and receivers, and would

need to be aware of the transmission format of the bursts; however no additional electronic routing or switching capability would be required. Delay lines may be acceptable in prototype switches, but are not commercially viable.

5.2 Wavelength Conversion

In WDM, several wavelengths run on a fiber link that connects two optical switches. The multiple wavelengths can be exploited to minimize contentions as follows. Let us assume that two bursts are destined to go out of the same output port at the same time. Both bursts can still be transmitted, but on two different wavelengths. This method may have some potential in minimizing burst contentions, particularly since the number of wavelengths that can be coupled together onto a single fiber continues to increase. For instance, it is expected there will be as many as 160-320 wavelengths per fiber in the near future.

Wavelength conversion is the process of converting the wavelength of an incoming channel to another wavelength at the outgoing channel. Wavelength converters are devices that convert an incoming signal's wavelength to a different outgoing wavelength, thereby increasing *wavelength reuse*, i.e., the same wavelength may be spatially reused to carry different connections in different fiber links in the network. Wavelength converters offer a 10%-40% increase in reuse values when wavelength availability is small [14].

In optical burst switching with wavelength conversion, contention is reduced by utilizing additional capacity in the form of multiple wavelengths per link [7, 1]. A contending burst may be switched to any of the available wavelengths on the outgoing link.

While optical wavelength conversion has been demonstrated in laboratory environments, the technology is not yet mature, and the range of possible conversions are somewhat limited [17]. The following are the different categories of wavelength conversion:

- *Full conversion:* Any incoming wavelength can be shifted to any outgoing wavelength; thus the is no wavelength continuity constraint on the end-to-end connection requests.

- *Limited conversion:* Wavelength shifting is restricted so that not all incoming channels can be connected to all outgoing channels. The restriction on the wavelength shifting will reduce the cost of the switch at the expense of increased blocking.

- *Fixed conversion:* This is a restricted form of limited conversion, wherein each incoming channel may be connected to one or more pre-determined outgoing channels.

- *Sparse wavelength conversion:* The networks may be comprised of a collection of nodes having full, limited, fixed, and no wavelength conversion. There are many wavelength conversion algorithms to minimize the number wavelength converters [18–20].

5.3 Deflection Routing

In deflection routing, contention is resolved by routing data to an output port other than the intended output port. Deflection routing is generally not favored in electronic packet-switched networks due to potential looping and out-of-sequence delivery of packets; however, it may be necessary to implement deflection in all-optical burst-switched networks, where buffer capacity is very limited. While deflection routing has been investigated for electronic and photonic packet-switched networks [21–23], there is currently very little work which applies deflection to optical burst-switched networks.

In [21], hot-potato routing is compared to store-and-forward routing in a ShuffleNet. [22] and [23] compare hot-potato and deflection routing in ShuffleNet and Manhattan Street Network topologies. Since both the ShuffleNet and Manhattan Street Network are two-connected (each node has an outgoing degree of two), the choice of the deflection output port is obvious. When the nodal degree is greater than two, a method must be developed to select the alternate outgoing link when a deflection occurs. In [24], deflection routing is studied in irregular mesh networks. Rather than choosing the deflection output port arbitrarily, priorities are assigned to each output port, and the ports are chosen in the prioritized order.

In deflection routing, a deflected packet or burst typically takes a longer route to its destination, leading to increased delay and a degradation of the signal quality. Furthermore, it is possible that the packet or burst may loop indefinitely within the network, adding to congestion. Mechanisms must be implemented to prevent excessive path lengths. Such mechanisms may include a maximum-hop counter, or a constrained set of deflection alternatives [25, 26]. In [25], deflection is studied together with optical buffering in irregular mesh networks with variable-length packets. The nodes at which deflection can occur, as well as the options for the deflection port, are limited in such a way as to prevent looping for the given network. A general methodology for selecting

loopless-deflection options in any arbitrary network is given in [26, 11, 27].

Another issue in deflecting bursts is maintaining the proper offset between the header and payload of a deflected burst. Since the deflected burst must traverse a greater number of hops than if the burst had not been deflected, there may be a point at which the initial offset time may not be sufficient for the header to be processed and for the switch to be reconfigured before the data burst arrives to the switch. In order to eliminate problems associated with insufficient offset time, a number of different policies may be implemented. One approach is simply to discard the burst if the offset time is insufficient. Counter and timer-based approaches may also be used to detect and limit the number of hops that a burst experiences. Buffering approaches using fiber delay lines (FDLs) may also be applied; however, such approaches increase the complexity of the optical layer.

5.4 Burst Segmentation

In existing optical burst switching approaches, when contention between two bursts cannot be resolved through other means, one of the bursts will be dropped in its entirety, even though the overlap between the two bursts may be minimal. For certain applications which have stringent delay requirements but relaxed packet loss requirements, it may be desirable to lose a few packets from a given burst rather than losing the entire burst altogether. In [28], the authors introduced a novel contention resolution technique, *burst segmentation*, which minimizes packet losses by partitioning the burst into segments and dropping only those segments which contend with another burst. A significant advantage of burst segmentation is that it allows bursts to be preempted by other bursts. This ability to preempt bursts enables the possibility of handling contentions in a prioritized manner.

In burst segmentation, the burst consists of a number of basic transport units called segments. Each segment consist of a segment header and a payload. The segment header contains fields for synchronization bits, error correction information, source and destination information, and the length of the segment in the case of variable length segments. The segment payload may carry any type of data, such as IP packets, ATM cells, or Ethernet frames (Fig. 5.1). When two bursts contend with one another in the optical burst-switched network, only those segments of one burst which overlap with the other burst will be dropped, as shown in Fig. 5.2. If the switching time is non-negligible, then additional segments may be lost when the switch is being reconfigured. In subsequent discussions, the burst which arrives to the first will be re-

ferred to as the *original* burst, and the burst which arrives later will be referred to as the *contending* burst.

In order to maintain data and format transparency, the optical layer need not be aware of the actual segment boundaries and segment payload data format. In this case, the optical layer is only aware of information such as the burst source and destination nodes, the burst offset time, the burst duration, and possibly the burst priority. This transparency may lead to sub-optimal decisions with regard to minimizing data loss, as individual segments may end up being split into two parts, resulting in complete data loss for those segments; however, by maintaining transparency, the optical layer (core) remains fairly simple, and no additional computational overhead will be required at each core node.

If the segment boundaries are transparent in the all-optical core, then the nodes at the network edge must be responsible for defining and processing segments electronically. Furthermore, the receiving node must be able to detect the start of each segment and identify whether or not the segment is intact; thus, some type of error detection or error correction overhead must be included in each segment. Additional clock and signaling information may need to be stored in each segment header in order for the egress receiver node to identify and recover data stored in each segment. One possible implementation of segmentation is to define a segment as an Ethernet frame. If each segment consists of an Ethernet frame, then detection and synchronization can be performed by using the preamble field in the Ethernet frame header, while errors and incomplete frames can be detected by using the CRC field in the Ethernet frame; thus, no further control overhead would be required in each segment other than the overhead already associated with an Ethernet frame.

If segments are not defined as Ethernet frames, then the choice of the segment length becomes a key system parameter. The segment can be either fixed or variable in length. If segments are fixed in length, synchronization at the receiver becomes easier; however, variable-length segments may be able to accommodate variable-length packets in a more efficient manner. The size of the segment also offers a trade off between the loss per contention and the amount of overhead per burst. Longer segments will result in a greater amount of data loss when segments are dropped during contention; however, longer segments will also result in less overhead per segment, as the ratio of the segment header length to the segment payload length will be lower. In this chapter, we assume that each segment is an Ethernet frame which contains a fixed-length packet, and we do not address the issue of finding the optimal segment size.

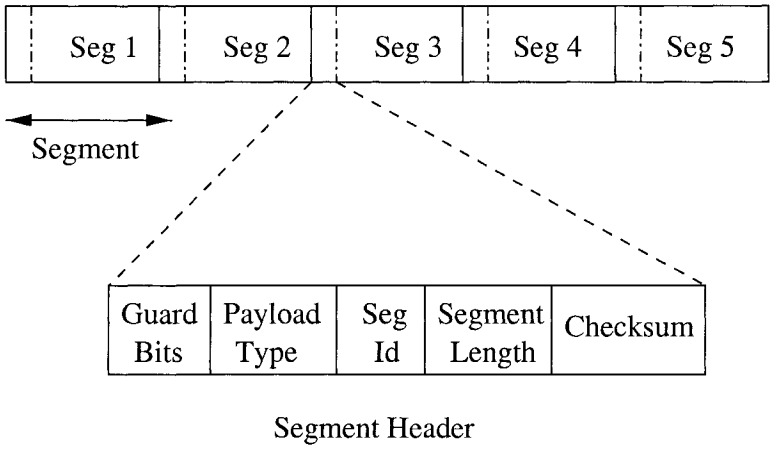


Figure 5.1. Segments header details.

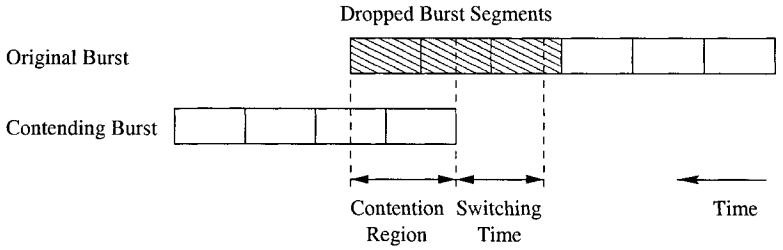


Figure 5.2. Selective segment dropping for two contending bursts.

Another issue in burst segmentation is the decision of which burst segments to drop when a contention occurs between two bursts. In the remainder of the dissertation, the burst arriving first to the switch is referred to as the *original burst* and the later arriving burst that contends is referred to as the *contending burst*. Note that the burst are referred to as original or contending burst based on the order of arrival of the data bursts to the switch, and not based on the order of arrival of their corresponding control packets (BHPs). There are two possible approaches for determining which segments to drop when using segmentation, namely, *tail-dropping* and *head dropping*. In tail-dropping, the overlapping tail segments of the original burst (Fig. 5.2) are dropped, and in head-dropping, the head overlapping segments of the contending burst are dropped. An advantage of dropping the overlapping tail segments of bursts rather than the overlapping head segments is that there is a better chance of in-sequence delivery of packets at the destination, assuming that dropped packets are retransmitted at a later time. A

head-dropping policy will result in a greater likelihood that packets will arrive at their destination out of order; however, the advantage of head-dropping is that it ensures that, once a burst arrives at a node without encountering contention, then the burst is guaranteed to complete its traversal of the node without preemption by later bursts.

In this chapter, we consider a *modified tail-dropping* policy when determining which segments to drop. In this policy, the overlapping tail (remaining length) of the original burst is dropped only if the number of segments in the overlapping tail is less than the total number of segments in (total length of) the contending burst. If the number of segments in the overlapping tail is greater than the number of segments in the contending burst, then the entire contending burst is dropped. This approach reduces the probability of a short burst preempting a longer burst and also minimizes the number of packets lost during contention.

One issue that arises when the tail of a burst is dropped is that the header for the burst, which may be forwarded before the segmentation occurs, will still contain the original burst length; therefore, downstream nodes may not know that the burst has been truncated. If downstream nodes are unaware of a burst's truncation, then it is possible that the previously truncated tail segments will contend with other bursts, even though these tail segments have already been dropped at a previous upstream node. These contentions may result in unnecessary packet loss.

If a tail-dropping policy is strictly maintained throughout the network, then the tail of the truncated burst will always be preempted in the case of a contention, and will never preempt segments of any other contending burst. However, for the case in which tail dropping is not strictly maintained, some action must be taken to avoid unnecessary packet losses. A simple solution is to have the truncating node generate and send out a *trailer*, or a trailing control message, to indicate to the downstream nodes along the path, when the truncated burst ends. The trailer is created electronically at the core switch where the contention is being resolved, and the time to create the trailer can be included in the offset time as being a part of the burst header processing time, δ, at each node. Note that the trailer is necessary only if the modified-tail dropping approach is adopted. If head-dropping is employed, then the header of the truncated burst may be updated immediately at the contention node. Also, if strict tail-dropping is employed, then the dropped tail segments will always lose the contention and will never preempt other segments, even at the downstream nodes along the path to the destination.

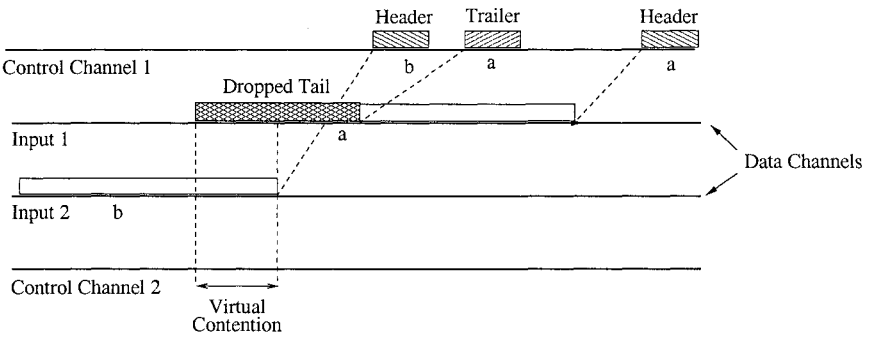


Figure 5.3. Trailer packet effective.

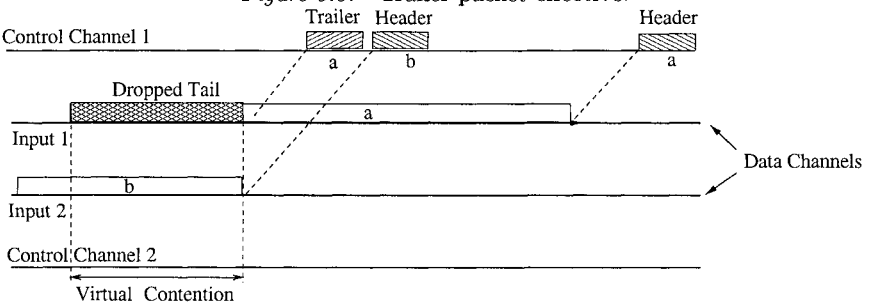


Figure 5.4. Trailer packet ineffective.

We note that, even if a trailer is created, the trailer may not be completely effective in eliminating contentions with burst segments that have already been dropped. Fig. 5.3 shows the situation in which the trailer packet reaches the downstream node before the header of a contending burst (Burst *b*). As soon as the trailer packet is received, the node is updated with the new length of the original burst (Burst *a*); hence, when the control header of the contending burst (Burst *b*) arrives, the virtual contention is avoided. In the case of Fig. 5.4, the header of the contending burst (Burst *b*) arrives before the trailer of the original burst (Burst *a*) at the downstream node; hence the switch detects a contention, even though the tail packets of the original burst have already been dropped. Although the trailer packet does not completely eliminate the situation of a virtual contention, as in the latter case, the trailer can minimize such situations; hence it is important to generate and transmit the trailer as soon as possible at the upstream node.

An additional system parameter which has a significant effect on burst segmentation is the switching time. If the node does not implement any buffering or other delaying mechanism, the switching time is a direct measure of the number of packets lost while reconfiguring

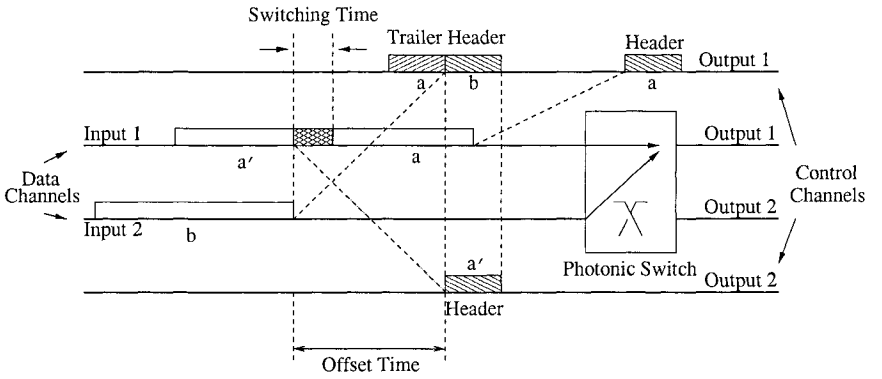


Figure 5.5. Segmentation with deflection policy for two contending bursts.

the switch due to a contention. Hence, a slow switching time will result in higher packet loss, while a fast switching time will result in lower packet loss. Current all-optical switches using micro-electro-mechanical systems (MEMS) [15, 30] technology are capable of switching on the order of milliseconds, while switches using semiconductor optical amplifier (SOA) technology are capable of switching on the order of nanoseconds. Due to their high switching times, MEMS switches may not by very suitable for optical burst switching, and are more appropriate for circuit-switched optical networks. On the other hand, SOA switches have been demonstrated in laboratory experiments [31], but have yet to be deployed in practical systems. In our experiments, we assume an intermediate and more practical switching time of 10 microseconds.

5.5 Segmentation with Deflection

A basic extension of burst segmentation is to implement segmentation with deflection. Rather than dropping one of the overlapping segments of a burst in contention, we can either deflect the entire contending burst or deflect the overlapping segments of the burst to an alternate output port other than the intended (original) output port. This approach is referred to as deflection routing or hot-potato routing [21, 23, 22]. Implementing segmentation with deflection (Fig. 5.5) increases the probability that the burst will reach the destination, and hence, may improve the performance. One problem which may arise is that a burst may encounter looping in the network or may be deflected multiple times, thereby wasting network bandwidth. This increased use of bandwidth can lead to increased contention and packet loss under high load conditions [25]. Due to deflection, the burst may also traverse a longer route, thereby increasing the total processing time. Deflection may also lead to

a situation in which the initial offset time is insufficient to transmit the data burst all-optically without storage. In order to avoid these problems, the burst will be dropped when the hop-count of the burst reaches a certain threshold [32–34].

When a burst is deflected, a deflection port must be selected. There may be one or many alternate deflection ports. The alternate deflection ports can either be determined ahead of time using a fixed port-assignment policy, which chooses the port based on the next shortest path, or determined dynamically using a load-balanced approach, which deflects the burst to an under-utilized link. In this chapter, we consider only one alternate deflection port, and choose the port which results in the second shortest path to the destination.

Selection of which burst (or burst-segments) to deflect during contention may be done in one of two ways. The first approach is to deflect the burst with the shorter remaining length (taking switching time into account). If the alternate port is busy, the burst may be dropped (Fig. 5.5). The second approach is to incorporate priorities into the burst. In this case, the lower-priority burst is deflected or segmented [35].

When combining segmentation with deflection, there are two basic approaches for ordering the contention resolution policies, namely, *segment-first* and *deflect-first*. In the segment-first policy, if the remaining length of the original burst is shorter than the contending burst, then the original burst is segmented and its tail is deflected. In case the alternate port is busy, the deflected part of the original burst is dropped. If the contending burst is shorter than the remaining length of the original burst, then the contending burst is deflected or dropped. In the deflect-first policy, the contending burst is deflected if the alternate port is free. If the alternate port is busy and if the remaining length of the original burst is shorter than the length of the contending burst, then the original burst is segmented and its tail is dropped. If the contending burst was found to be shorter, then the contending burst is dropped.

An example of the segmentation-deflection scheme is shown in Fig. 5.5. Initially when the header for Burst a arrives at the switch, it is routed onto Output Port 1. Once the header of Burst b arrives at the switch, there is a contention. Since the offset time is common to all of the bursts, the header indicates when and where the bursts will contend. Therefore, by taking the switching time into consideration, and by using the segment-first policy, one of the bursts will deflected (or segmented and deflected) to the alternate port if the alternate port is free and will be dropped if the alternate port is not free. Here, the remaining length of Burst a is less than the length of Burst b. Hence, Burst a is segmented

and its tail is deflected to the alternate port as a new burst. A header is created for the deflected new burst and is sent on Output Port 2. This new header is generated at the time that the header of Burst b is processed. A trailer is created for the segmented Burst a and is sent on the control channel of Output Port 1. Packets of the segmented burst are lost during the reconfiguration of the switch. In the policy that utilizes both segmentation and deflection, the processing time δ at each node includes the time to create a header for the new burst segment in the case of a contention; hence the offset time remains the same as in the case of standard optical burst switching.

A possible side-effect of segmentation with deflection is that, when there is a contention, the shorter remaining burst will be segmented and will be deflected as a new burst. Creating these new bursts may lead to burst fragmentation, in which there are many short bursts propagating through the network. These short bursts will incur higher overhead with respect to switching times and control overhead per burst. Furthermore, having a greater number of smaller bursts in the network will also increase the number of control packets. These additional control packets may overload the control plane; hence, it may be advisable to drop the segmented burst if the new burst length is lower than a minimum burst size.

Fragmentation may be alleviated by utilizing the modified tail-dropping policy. In the modified tail-dropping policy, the lengths of the two contending bursts are compared and the smaller of the contending burst or the remaining part of the original burst is deflected or segmented, respectively. If a deflection port is unavailable, then the segments that lose the contention will be dropped. Thus, the short, fragmented bursts are more likely to be dropped, and will not significantly hinder other bursts.

Another issue in deflecting bursts is maintaining the proper offset between the header and payload of a deflected burst. Since the deflected burst must traverse a greater number of hops than if the burst had not been deflected, there may be a point at which the initial offset time may not be sufficient for the header to be processed and for the switch to be reconfigured before the data burst arrives to the switch. In order to eliminate problems associated with insufficient offset time, a number of different policies may be implemented. One approach is simply to discard the burst if the offset time is insufficient. Counter and timer-based approaches may also be used to detect and limit the number of hops that a burst experiences. If the goal is to minimize packet loss, then the head of the burst can simply be truncated while a switch is being configured, and the tail segments of the burst can continue through the

network. Buffering approaches using fiber delay lines (FDLs) may also be applied; however, such approaches increase the complexity of the optical layer.

Another issue when implementing segmentation and deflection is how to handle long bursts which may span multiple nodes simultaneously. If a long burst passing through two or more switches experiences contention from two or more different bursts at different switches, then, based on the timing of these contentions, the contentions may be resolved in a number of ways. If an upstream node segments the burst first, then the downstream nodes are updated by the trailer packet to eliminate unnecessary contentions. On the other hand, if the contention occurs at the downstream node before the upstream node, and if the burst's tail is deflected at the downstream node, then the upstream contentions will not be affected. If the downstream node drops the tail of the burst, then the upstream node will not know about the truncation and will continue to transmit the tail. The downstream node may send a control message to the upstream node in order to reduce unnecessary contentions with the tail at the upstream node. In the case where more than two bursts contend at the same switch, the contention is handled sequentially.

One possible advantage of segmentation in optical burst-switched networks is that it can provide an additional degree of differentiation for supporting different quality of service (QoS) requirements. When two bursts contend with one another, the burst priority can be used to determine which burst to segment or drop. For example, if a high priority burst arrives to a node and finds that a low priority burst is being transmitted on the desired output, then the low priority burst can be segmented, and its tail can be dropped in order to transmit the high priority burst. On the other hand, if a low priority burst arrives to a node and finds a high priority burst being transmitted, then the low priority burst will be dropped. When combining segmentation with deflection, an even greater degree of differentiation may be achieved. The choice of whether to deflect the newly arriving contending burst, or the tail of the burst currently being transmitted, can be made based on priorities. Segmentation-based QoS schemes are studied in-detail in Chapter 7.

We evaluate the following five different policies for handling contention in the OBS network:

1. *Drop Policy (DP):* Drop the entire contending burst.

2. *Deflect and Drop Policy (DDP):* Deflect the contending burst to the alternate port. If the port is busy, drop the burst.

3. *Segment and Drop Policy (SDP):* The contending burst wins the contention. The original burst is segmented, and its segmented tail is dropped.

4. *Segment, Deflect and Drop Policy (SDDP):* The original burst is segmented, and its segmented tail may be deflected if an alternate port is free, otherwise the tail is dropped.

5. *Deflect, Segment and Drop Policy (DSDP):* The contending burst is deflected to a free port if available, otherwise the original burst is segmented and its tail is dropped, while the contending burst is transmitted.

In order to evaluate the performance of the segmentation and deflection schemes, we develop a simulation model. Fig. 5.6 shows the 14-node NSF network on which the simulation and analytical results are applied. The link distances are shown in km.

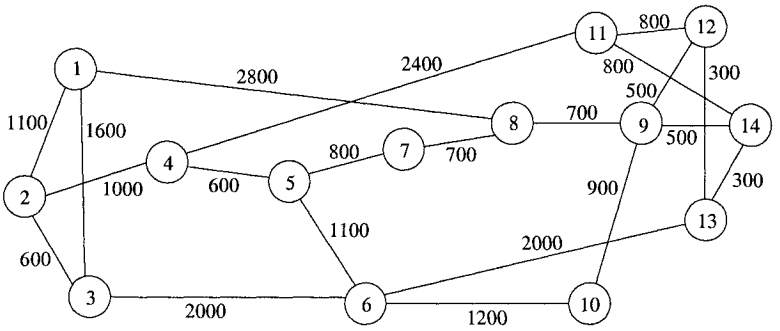


Figure 5.6. NSF network with 14 nodes (distances in km).

The following are the important assumptions in the simulation:

- Burst arrivals to the network are Poisson.
- Burst length is an exponentially generated random number rounded to the nearest integer multiple of the fixed packet length, with an average burst length of 100 μs.
- Transmission rate is 10 Gb/s.
- Packet length is 1500 bytes.
- Switching time is 10 μs.
- There is no buffering or wavelength conversion at nodes.

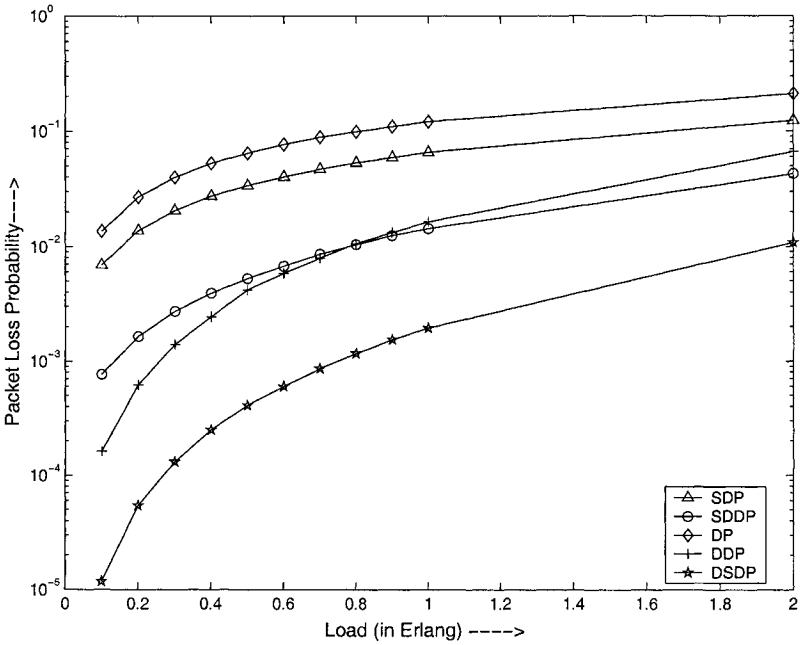


Figure 5.7. Packet loss probability versus load for NSFNET at low loads with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

- Traffic is uniformly distributed over all source-destination pairs.
- Fixed shortest path routing is used between all node pairs.

Figure 5.7 plots the total packet loss probability versus the load for the different contention resolution policies. An average burst length of 100 μs is assumed. We observe that SDP performs better than DP at all load conditions, and that the three policies with deflection, namely DSDP, SDDP, and DDP, perform better than the corresponding policies without deflection at low loads. DSDP performs better than SDDP and DDP at these loads; thus, at low loads, it is better to attempt deflection before segmentation. Also, at low loads DDP performs better than SDDP since there is no loss due to switching time in DDP. We see that policies with segmentation perform better than the policies without segmentation. A logical explanation would be that, in segmentation, on average only half of the packets from one of the bursts are lost when contention occurs (due to the exponential burst length assumption). Also, at low loads, there is a greater amount of spare capacity, increasing the chance of successful deflection.

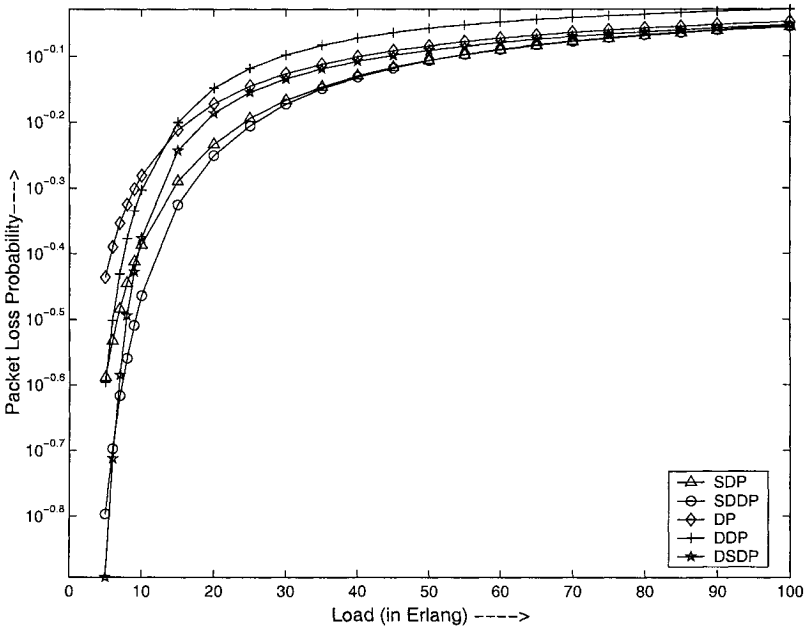


Figure 5.8. Packet loss probability versus load for NSFNET at high loads with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

Figure 5.8 shows the packet-loss performance at very high loads. DSDP performs the best only at low loads. SDDP performs the best when the total load into the network is between 6 and 55 Erlang, after which SDP performs equally well, if not better. DDP performs well only at low loads, while at very high loads DP fares better than DDP. We observe that, at very high loads, policies without deflection perform better then the policies with deflection. At high loads, deflection may add to the load, increasing the probability of contention, and thereby increasing loss.

Figure 5.9 shows the average number of hops versus load for the different policies. In the deflection policies, the number of deflections increases as the load increases, resulting in higher average hop distance at low loads. As the load increases further, those bursts which are further from their destination will experience more contention than those bursts which are close to their destination. Thus, bursts with higher average hop count are less likely to reach their intended destination, and the average hop distance will decrease as load increases. Policies with segmentation have higher hop count compared to their corresponding policies without segmentation, since the probability of a burst reaching its destination is higher with segmentation.

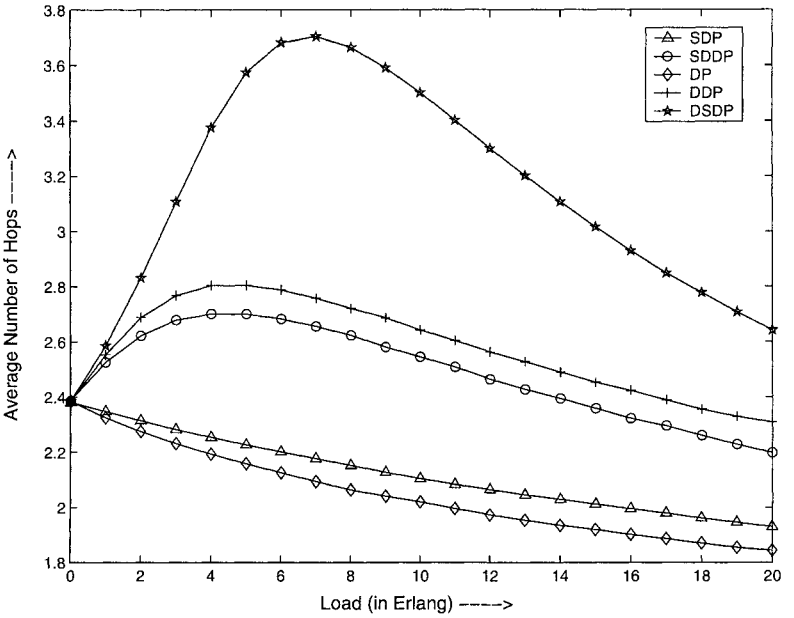


Figure 5.9. Average number of hops versus load for NSFNET with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

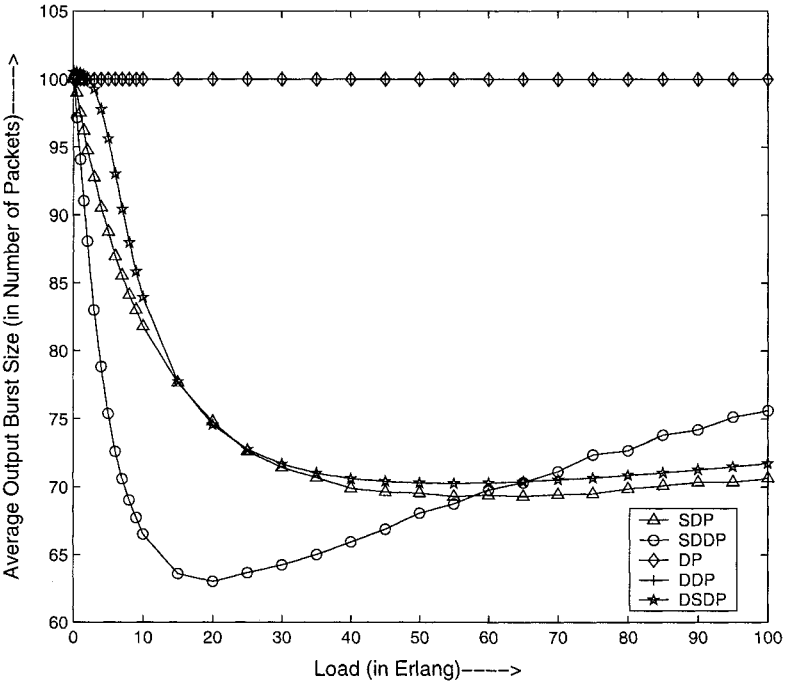


Figure 5.10. Average output burst size versus load for NSFNET with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

Figure 5.10 shows the average output burst size versus load for the different policies. The output burst size is measured over both dropped and successfully received bursts. Initially, the burst size decreases with increasing load, as there are more segmentations with the increasing number of contentions. As the load increases further, the segmented bursts encounter more contentions, and because the segmented bursts have smaller size (lower priority), the segmented bursts tend to be dropped. The values for DP and DDP are constant for different values of load because the size of a burst is never altered.

The packet loss probability versus load for different values of switching time is shown in Fig. 5.11. As the switching time increases, the performance of SDDP decreases because a greater number of packets are lost during the re-configuration of the switch. On the other hand, DDP is not affected by the switching time and the loss remains almost constant. At low switching times, the results show that SDDP is better than the standard DDP, while at higher switching times, the standard DDP is better than the new SDDP because of the loss of packets during the switching time.

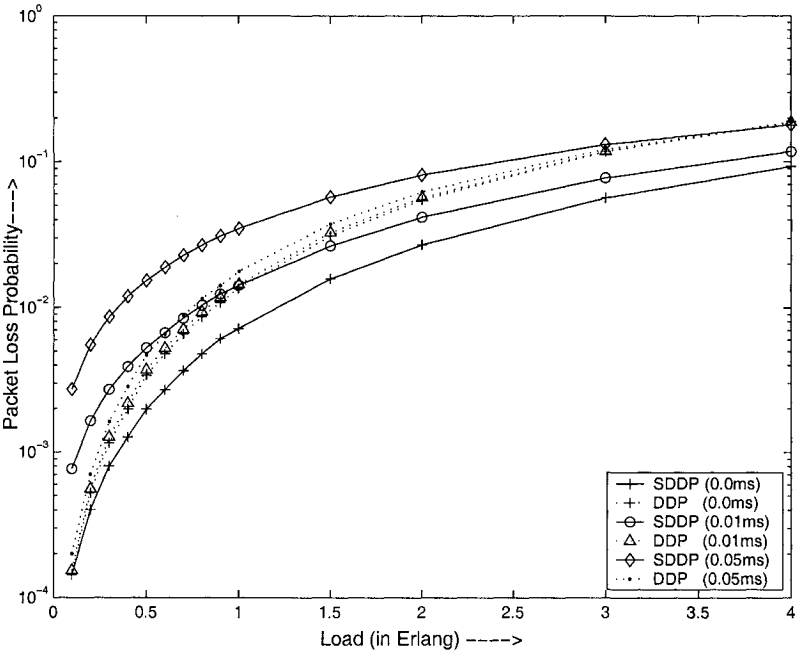


Figure 5.11. Packet loss probability versus load at varying switching times for NSFNET with $\frac{1}{\mu} = 100\mu$s and Poisson burst arrivals.

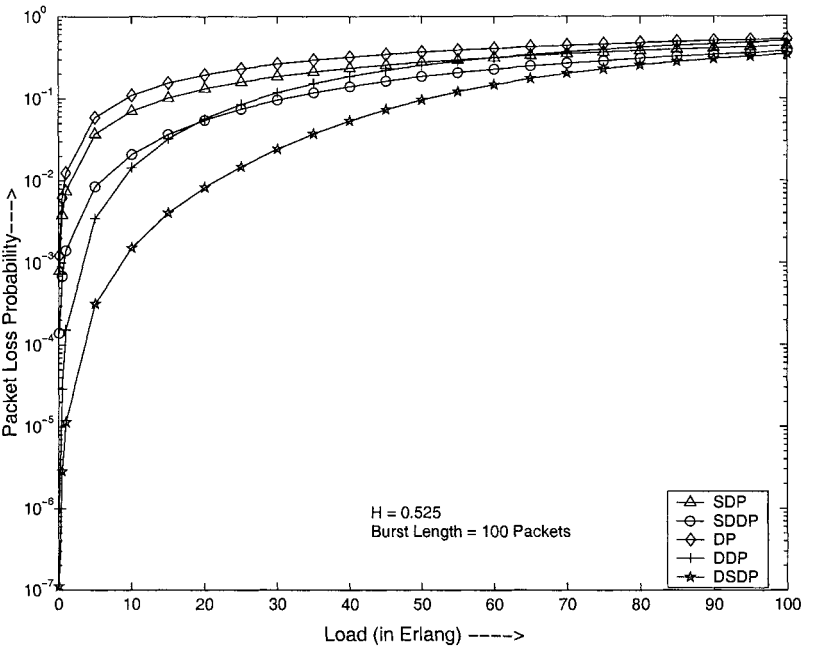


Figure 5.12. Packet loss probability versus load for NSFNET with Pareto burst arrivals.

In order to capture the burstiness of data at the edge nodes, we also simulate Pareto burst arrivals with 100 independent traffic sources. The length of the burst is fixed to the average burst length in the Poisson case, i.e., 100 fixed-sized packets. The Hurst parameter, H is set to 0.525. The remaining assumptions are the same. We plot the graphs for packet loss probability, average hop count, and output burst size versus load for Pareto inter-arrival time distribution and fixed-sized bursts.

Figure 5.12 plots the total packet loss probability versus the load with Pareto burst arrivals, for the different contention resolution policies. The results are similar to the Poisson case, except that DSDP is the best policy for the observed load range. We also observe that the policies with deflection perform better than the Poisson case due to the increased burstiness at the source. Deflection is a good option to avoid the contentions at the source.

Figure 5.13 shows the average number of hops versus load with Pareto burst arrivals for the policies. Figure 5.14 shows the average output burst size versus load with Pareto burst arrivals, for the different policies. The results are similar to the Poisson case.

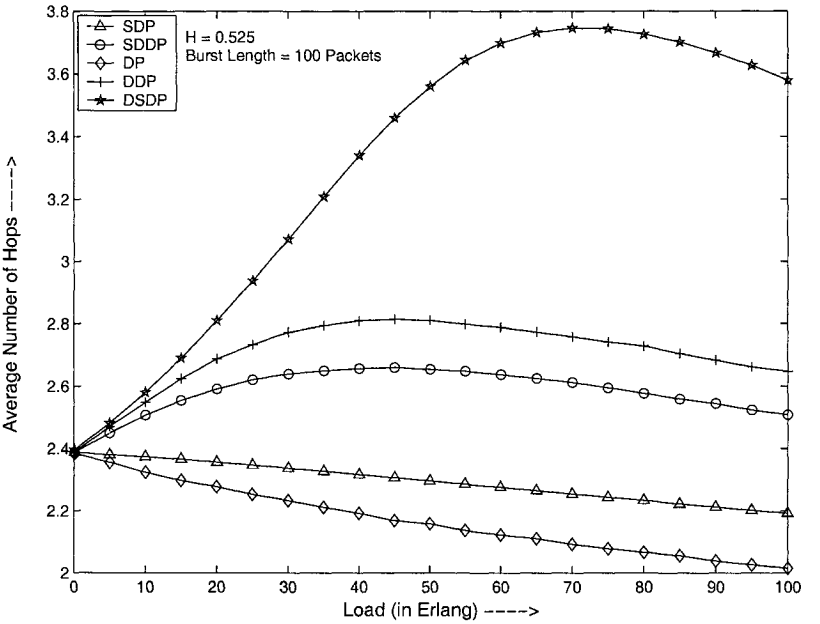


Figure 5.13. Average number of hops versus load for NSFNET with Pareto burst arrivals.

5.6 Contention Resolution and QoS

Contention resolution schemes may be used to provide QoS in an all-optical core network. In [8], an approach is introduced in which low-priority bursts are intentionally dropped under certain conditions in order to reduce loss for high-priority bursts. The scheme provides a proportional reduction rather than a complete elimination of high-priority burst losses due to contention with low-priority bursts. A limitation of the scheme is that it can result in the unnecessary dropping of low-priority bursts.

In [37], a priority-based deflection scheme is used to resolve contention in a photonic packet-switched network. Packets are assigned priorities, and the priorities are used to determine which packet to deflect or drop when a contention occurs. In [10], the authors have introduced a similar scheme for optical burst-switched networks. The scheme utilizes deflection as well as burst segmentation to resolve contentions. The results show a fairly significant differentiation between different burst priorities in terms of both packet loss and delay. Furthermore, the loss of packets in a high-priority burst due to contention with a low-priority bursts can be completely eliminated (100% class isolation).

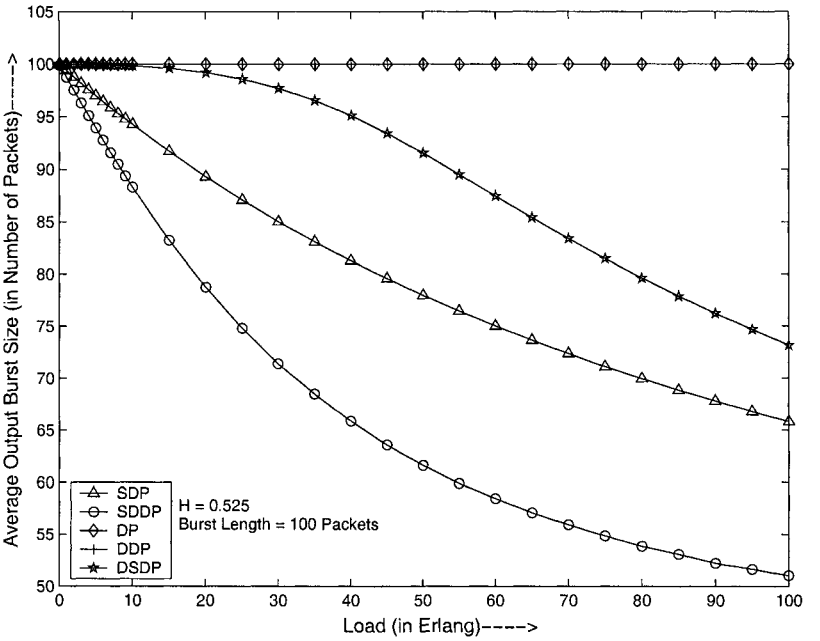


Figure 5.14. Average output burst size versus load for NSFNET with Pareto burst arrivals.

References

[1] D. K. Hunter, M. C. Chia, and I. Andonovic. Buffering in optical packet switches. *IEEE/OSA Journal of Lightwave Technology*, 16(12):2081–2094, December 1998.

[2] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, and I. Andonovic. SLOB: A switch with large optical buffers for packet switching. *IEEE/OSA Journal of Lightwave Technology*, 16(10):1725–1736, October 1998.

[3] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and et al. CORD: Contention resolution by delay lines. *IEEE Journal on Selected Areas in Communications*, 14(5):1014–1029, June 1996.

[4] Z. Haas. The 'Staggering Switch': An electronically controlled optical packet switch. *IEEE/OSA Journal of Lightwave Technology*, 11(5/6):925–936, May/June 1993.

[5] I. Chlamtac, A. Fumagalli, and C.-J. Suh. Multibuffer delay line architectures for efficient contention resolution in optical switching nodes. *IEEE Transactions on Communications*, 48(12):2089–2098, December 2000.

[6] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, and et al. SLOB: A switch with large optical buffers for packet switching. *IEEE/OSA Journal of Lightwave Technology*, 16(10):1725–1736, October 1998.

[7] L. Tancevski, G. Castanon, F. Callegati, and L. Tamil. Performance of an optical IP router using non-degenerate buffers. In *Proceedings, IEEE Globecom*, pages 1454–1459, December 1999.

[8] G. Bendeli and et al. Performance assessment of a photonic atm switch based on a wavelength controlled fiber loop buffer. In *Proceedings, Optical Fiber Communication Conference (OFC)*, pages 106–107, 1996.

[9] W. D. Zhong and R. S. Tucker. Wavelength routing based photonic packet buffers and their applications in photonic packet switching systems. *IEEE/OSA Journal of Lightwave Technology*, 16(10):1737–1745, October 1998.

[10] M. C. Chia, D. K. Hunter, I. Andonovic, P. Ball, I. Wright, S. P. Ferguson, K. M. Guild, and M. J. O'Mahony. Packet loss and delay performance of feedback and feed-forward arrayed-waveguide gratings-based optical packet switches with WDM inputs-outputs. *IEEE/OSA Journal of Lightwave Technology*, 19(9):1241–1254, September 2001.

[11] T. Zhang, K. Lu, and J. P. Jue. Differentiated contention resolution for QoS in photonic packet-switched networks. In *Proceedings, IEEE International Conference on Communications (ICC)*, June 2004.

[12] S. Yao, B. Mukherjee, and S. Dixit. Advances in photonic packet switching: An overview. *IEEE Communications Magazine*, 38(2):84–94, February 2000.

[13] F. Callegati. Optical buffers for variable length packets. *IEEE Communications Letters*, 4(9):292–294, September 2000.

[14] R. Ramaswami and K.N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transactions on Networking*, 3(5):489–500, October 1995.

[15] M. Yoo, C. Qiao, and S. Dixit. QoS performance of optical burst switching in IP-over-WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2062–2071, October 2000.

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

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

[18] G. Xiao and Y. Leung. Algorithms for allocating wavelength converters in all-optical networks. *IEEE/ACM Transactions on Networking*, 7(4):545–557, August 1999.

[19] H. Zang, J.P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *SPIE Optical Networks Magazine*, 1(1), January 2000.

[20] S. L. Danielsen and et al. Analysis of a WDM packet switch with improved performance under bursty traffic conditions due to tunable wavelength converters. *IEEE/OSA Journal of Lightwave Technology*, 16(5):729–735, May 1998.

[21] A. S. Acampora and I. A. Shah. Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing. *IEEE Transactions on Communications*, 40(6):1082–1090, June 1992.

[22] F. Forghieri, A. Bononi, and P. R. Prucnal. Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks. *IEEE Transactions on Communications*, 43(1):88–98, January 1995.

[23] A. Bononi, G. A. Castanon, and O. K. Tonguz. Analysis of hot-potato optical networks with wavelength conversion. *IEEE/OSA Journal of Lightwave Technology*, 17(4):525–534, April 1999.

[24] G. Castanon, L. Tancevski, and L. Tamil. Routing in all-optical packet switched irregular mesh networks. In *Proceedings, IEEE Globecom*, pages 1017–1022, December 1999.

[25] S. Yao, B. Mukherjee, S.J.B. Yoo, and S. Dixit. All-optical packet-switched networks: A study of contention resolution schemes in an irregular mesh network with variable-sized packets. In *Proceedings, SPIE OptiComm*, pages 235–246, October 2000.

[26] J.P. Jue. An algorithm for loopless deflection in photonic packet-switched networks. In *Proceedings, IEEE International Conference on Communications (ICC)*, April 2002.

[27] T. Zhang, K. Lu, and J. P. Jue. Differentiated contention resolution for QoS in photonic packet-switched networks. *IEEE/OSA Journal of Lightwave Technology*, 2004.

[28] V.M. Vokkarane, J.P. Jue, and S. Sitaraman. Burst segmentation: an approach for reducing packet loss in optical burst switched networks. In *Proceedings, IEEE International Conference on Communications (ICC)*, volume 5, pages 2673–2677, April 2002.

[29] A. Neukermans and R. Ramaswami. Mems technology for optical networking applications. *IEEE Communications Magazine*, 39(1):62–69, January 2001.

[30] T.-W. Yeow, K.L.E. Law, and A. Goldenberg. Mems optical switches. *IEEE Communications Magazine*, 39(11):158–163, November 2001.

[31] R. Ramaswami and K.N.Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, ch. 3, 126-160, 1998.

[32] X. Wang, H. Morikawa, and T. Aoyama. Burst optical deflection routing protocol for wavelength routing WDM networks. In *Proceedings, SPIE OptiComm*, pages 257–266, 2000.

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

[34] S. Lee, K. Sriram, H. Kim, and J. Song. Contention-based limited deflection routing in OBS networks. In *Proceedings, IEEE Globecom*, pages 2633–2637, December 2003.

[35] S. Kim, N. Kim, , and M. Kang. Contention resolution for optical burst switching networks using alternative routing. In *Proceedings, IEEE International Conference on Communications (ICC)*, volume 5, pages 2678–2681, May 2002.

[36] Y. Chen, M. Hamdi, and D.H.K. Tsang. Proportional QoS over OBS network. In *Proceedings, IEEE Globecom*, volume 3, pages 1510–1514, November 2001.

[37] S. Yao, S.J.B. Yoo, and B. Mukherjee. A comparison study between slotted and unslotted all-optical packet-switched networks with priority-based routing. In *Proceedings, Optical Fiber Communication Conference (OFC)*, March 2001.

[38] V. M. Vokkarane and J. P. Jue. Prioritized routing and burst segmentation for QoS in optical burst-switched networks. In *Proceedings, Optical Fiber Communication Conference (OFC)*, volume WG6, pages 221–222, March 2002.

Chapter 6

CHANNEL SCHEDULING

When a burst arrives to a node, it must be assigned a wavelength on the appropriate outgoing link. In this problem, all-optical wavelength conversion is assumed to be available at each node, and the scheduling occurs at intermediate core nodes as well as ingress nodes. The primary objective in this type of scheduling is to minimize the "gaps" in each channel's schedule, where a gap is the idle space between two bursts which are transmitted over the same output wavelength. Channel scheduling in OBS networks is different from traditional IP scheduling, since, in IP, each core node stores the packets in electronic buffers and schedules them on the desired output port. In OBS, once a burst arrives at a core node, it must be sent to the next node without storing the burst in electronic buffers. We assume that each OBS core node supports full-optical wavelength conversion.

When a BHP arrives at a core node, a channel scheduling algorithm is invoked to assign the unscheduled burst to a data channel on the outgoing link. The channel scheduler obtains the burst arrival time and duration of the unscheduled burst from the BHP. The algorithm may need to maintain the latest available unscheduled time (LAUT) or the horizon, gaps, and voids on every outgoing data channel. Traditionally, the LAUT of a data channel is the earliest time at which the data channel is available for an unscheduled data burst to be scheduled. A gap is the time difference between the arrival of the unscheduled burst and ending time of the previously scheduled burst. A void is the unscheduled duration (idle period) between two scheduled bursts on a data channel. For void filling algorithms, the starting and the ending time for each burst on every data channel must also be maintained.

The following information is used by the scheduler for most of the scheduling algorithms:

- L_b: Unscheduled burst length duration.
- t_{ub}: Unscheduled burst arrival time.
- W: Maximum number of outgoing data channels.
- N_b: Maximum number of data bursts scheduled on a data channel.
- D_i: i^{th} outgoing data channel.
- $LAUT_i$: LAUT of the i^{th} data channel, $i = 1, 2, ..., W$, for non-void filling scheduling algorithms.
- $S_{(i,j)}$ and $E_{(i,j)}$: Starting and ending times of each scheduled burst, j, on every data channel, i, for void filling scheduling algorithms.
- Gap_i: If the channel is available, gap is the difference between t_{ub} and $LAUT_i$ for scheduling algorithms without void filling, and is the difference between t_{ub} and $E_{(i,j)}$ of previous scheduled burst, j, for scheduling algorithms with void filling. If the channel is busy, Gap_i is set to 0. Gap information is useful to select a channel for the case in which more than one channel is free.

Data channel scheduling algorithms can be broadly classified into two categories: with and without void filling. The algorithms primarily differ based on the type and amount of state information that is maintained at a node about every channel. In data channel scheduling algorithms without void filling, the $LAUT_i$ on every data channel D_i, $i = 0, 1, ..., W$, is maintained by the channel scheduler. In void filling algorithms, the starting time, $S_{(i,j)}$ and ending time, $E_{(i,j)}$ are maintained for each burst on every data channel, where, $i = 0, 1, ..., W$, is the i^{th} data channel and $j = 0, 1, ..., N_b$, is the j^{th} burst on channel i.

Let the initial data channel assignment for the channel scheduling algorithms without void filling and with void filling be as shown in Fig. 6.1(a) and (b), respectively. In Fig. 6.1(a), the $LAUT_i$ on every data channel D_i, $i = 0, 1, ..., W$, is maintained by the scheduler. In Fig. 6.1(b), the starting time, $S_{(i,j)}$ and the ending time, $E_{(i,j)}$, where i refer to the i^{th} data channel and j is the j^{th} burst on channel i, are maintained for each burst on every output data channel. In the following subsections, we will describe traditional non-void filling scheduling algorithms, such as First Fit Unscheduled Channel (FFUC) and Latest Available Unscheduled Channel (LAUC), and traditional void-filling scheduling algorithms, such as First Fit Unscheduled Channel with Void

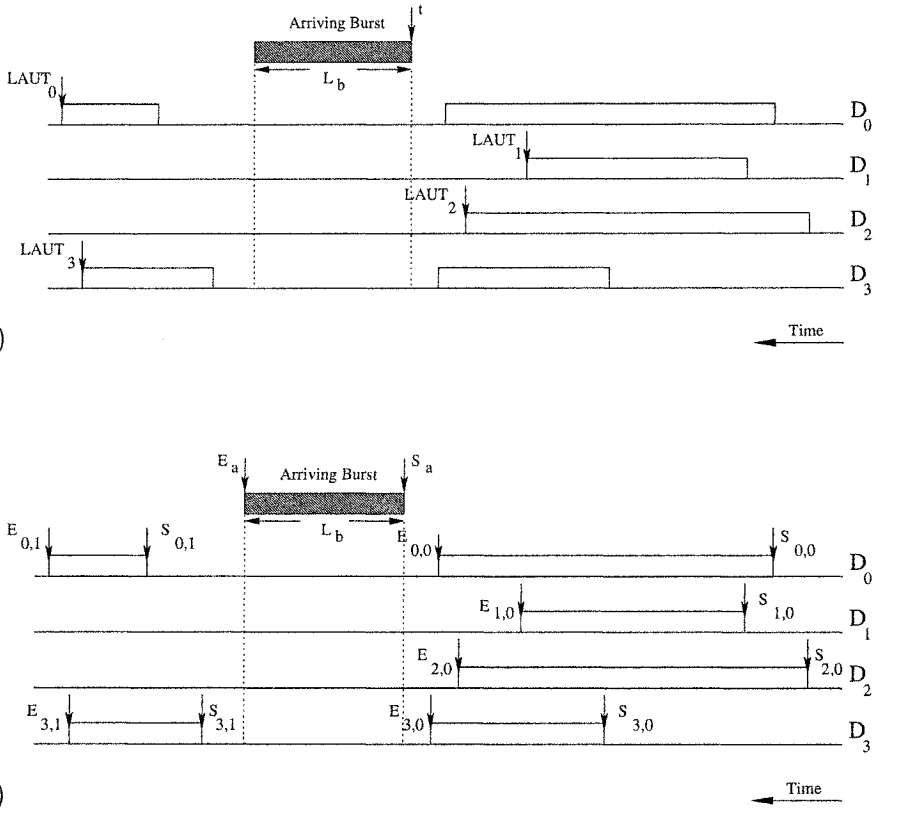


Figure 6.1. Initial data channel status (a) without void filling (b) with void filling.

Filling(FFUC-VF) and Latest Available Unscheduled Channel with Void Filling (LAUC-VF).

First Fit Unscheduled Channel (FFUC):

The FFUC scheduling algorithm keeps track of the LAUT (or horizon) on every data channel. A wavelength is considered for each arriving burst when the unscheduled time (LAUT) of the data channel is less then the burst arrival time. The FFUC algorithm searches all the channels in a fixed order and assigns the first available channel for the new arriving burst. The primary advantage of FFUC is the simplicity of the algorithm and that the algorithm needs to maintain only one value ($LAUT_i$) for each channel. The FFUC algorithm can be illustrated in Fig. 6.2(a). Based on the $LAUT_i$, data channels D_1 and D_2 are available for the duration of the unscheduled burst. If the channels are ordered based on the index of the wavelengths (D_0, D_1, ..., D_W), the arriving burst is scheduled on outgoing data channel D_1. The time complexity of the

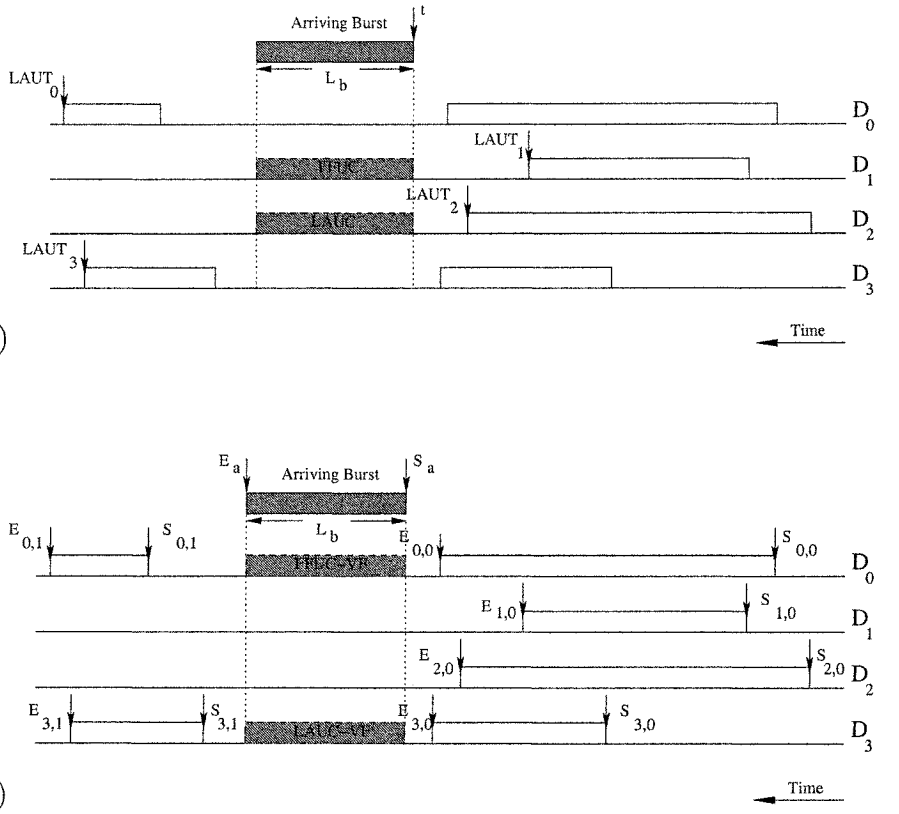


Figure 6.2. Channel assignment after using (a) non void filling algorithms (FFUC and LAUC), and (b) void filling algorithms (FFUC-VF and LAUC-VF).

FFUC algorithm is $O(\log W)$. The primary drawback of FFUC is the high burst dropping probability as a trade-off for simplicity in scheduling. The following algorithms aim at reducing the burst dropping probability at the expense of increased algorithm complexity.

Horizon or Latest Available Unscheduled Channel (LAUC): The LAUC or Horizon [1] scheduling algorithm keeps track of the LAUT (or horizon) on every data channel and assigns the data burst to the latest available unscheduled data channel. The LAUC algorithm can be illustrated in Fig. 6.1(a). Based on the $LAUT_i$, data channels D_1 and D_2 are available for the duration of the unscheduled burst. Also, we observe that $Gap_1 > Gap_2$; thus, the arriving burst is scheduled on outgoing data channel with the minimum gap, i.e., D_2. The time complexity of the LAUC algorithm is $O(\log W)$.

First Fit Unscheduled Channel with Void Filling (FFUC-VF):

The FFUC-VF scheduling algorithm maintains the starting and ending times for each scheduled data burst on every data channel. The goal of this algorithm is to utilize voids between two data burst assignments. The first channel with a suitable void is chosen. The FFUC-VF algorithm is illustrated on Fig. 6.1(b). Based on the $S_{i,j}$ and $E_{i,j}$, all the data channels D_0, D_1, D_2, and D_3 are available for the duration of the unscheduled burst. If the channels are ordered based on the index of the wavelengths (D_0, D_1, ..., D_W), the arriving burst is scheduled on outgoing data channel D_0. If N_b is the number of bursts currently scheduled on every data channel, then a binary search algorithm can be used to check if a data channel is eligible. Thus, the time complexity of the LAUC-VF algorithm is $O(\log(WN_b))$.

Latest Available Unscheduled Channel with Void Filling (LAUC-VF):

The LAUC-VF [2] scheduling algorithm maintains the starting and ending times for each scheduled data burst on every data channel. The goal of this algorithm is to utilize voids between two data burst assignments. The channel with a void that minimizes the gap is chosen. The LAUC-VF algorithm is illustrated on Fig. 6.1(b). Based on the $S_{i,j}$ and $E_{i,j}$, all the data channels D_0, D_1, D_2, and D_3 are available for the duration of the unscheduled burst. Also, we observe that D_3 had the least gap Gap_3; thus, the arriving burst is scheduled on D_3. If N_b is the number of bursts currently scheduled on every data channel, then a binary search algorithm can be used to check if a data channel is eligible. Thus, the time complexity of the LAUC-VF algorithm is $O(\log(WN_b))$.

Recently, researcher have proposed several optimizations for the above described scheduling algorithms. In [3], a *Minimizing Voids Unscheduled Channel (MVUC)* algorithm proposes with the objective of minimizing voids generated by arriving bursts at each core node. In the proposed scheduling algorithm, when the burst which has arrived at optical core router at a certain time can be transmitted in some data channels by using the unused data channel capacity, the MVUC algorithm selects the data channel in which the newly generated void after scheduling the arriving burst becomes minimum. The authors conclude through computer simulations that the MVUC performs better than LAUC-VF in terms data loss.

[4] proposes the *Minimum Starting Void (Min-SV)* algorithm for selecting channels for incoming data bursts. The advantage of Min-SV is that it has the same scheduling criteria as LAUC-VF. However, the data structure of Min-SV is constructed by augmenting a balanced binary search tree. By constructing this tree, Min-SV achieves a loss rate as low as LAUC-VF and processing time as low as Horizon (LAUC).

The Look-ahead Window (LAW) [5] or a Group-based Scheduling algorithm [6], takes advantage of the separation between the data bursts and the burst header packets (offset time). By receiving BHPs one offset time prior to their corresponding data bursts, it is possible to construct a lookahead window. The authors believe that such a collective view of multiple BHPs results in more efficient decisions with regard to which incoming bursts should be discarded or reserved. Also, the use of FDLs for any lost time in the offset, due to the creating of a window is suggested.

There has also been substantial work on scheduling using FDLs in OBS [13–1, 8]. In the following sections, we described several scheduling algorithms that are based on burst segmentation [9], with and without FDLs. We shown that these algorithms can achieve significantly lower loss than all the above scheduling algorithms [10, 11].

6.1 Segmentation-Based Channel Scheduling

In this chapter, we consider an OBS network where each WDM link consists of *control channels* used to transmit BHPs, and *data channels* used to transmit data bursts. We also assume that every channel consists of a wavelength and that each OBS core router has wavelength conversion capability. We address the important issue of scheduling data bursts onto outgoing data channels at every OBS core router.

When a BHP arrives at a node, a channel scheduling algorithm is invoked to assign the unscheduled burst to a data channel on the outgoing link. The channel scheduler obtains the burst arrival time and duration of the unscheduled burst from the BHP. The algorithm may need to maintain the latest available unscheduled time (LAUT) or the horizon, gaps, and voids on every outgoing data channel. Traditionally, the LAUT of a data channel is the earliest time at which the data channel is available for an unscheduled data burst to be scheduled. A gap is the time difference between the arrival of the unscheduled burst and ending time of the previously scheduled burst. A void is the unscheduled duration between two scheduled bursts on a data channel. For void filling algorithms, the starting and the ending time for each burst on every data channel must also be maintained.

The scheduling algorithm must find an available data channel on the appropriate output port for each incoming burst in a manner which is quick and efficient, and which minimizes data loss. In order to minimize data loss, the scheduling algorithm may use one or more contention resolution techniques. Traditional data channel scheduling algorithms are classified into two categories, namely non-void filling algorithms and void-filling algorithms. Non-void filling algorithms include first fit

unscheduled channel (FFUC) and latest available unscheduled channel (LAUC)[1]. Void filling algorithms include first fit unscheduled channel with void filling (FFUC-VF) and latest available unscheduled channel with void filling (LAUC-VF) [13]. The performance of scheduling algorithms can be enhanced by using optical buffering (FDLs), wavelength converters, and deflection routing techniques for resolving burst contentions [1, 13, 8, 12–14]. However, these contention resolution techniques drop the burst completely if they fail to resolve the contention. Instead of dropping the burst in its entirety, it is possible to drop only the overlapping parts of a burst using the burst segmentation technique.

Due to the inherent property of segmentation, the segmentation-based channel scheduling algorithms can be either *non-preemptive* or *preemptive.* In the non-preemptive approach, existing channel assignments are not altered, while in preemptive scheduling algorithms, an arriving unscheduled burst [1] may preempt existing data channel assignments, and the preempted bursts (or burst segments) may be rescheduled or dropped.

The advantage of a non-preemptive approach is that the BHP of the segmented unscheduled burst can be immediately updated with the corresponding change in the burst length and arrival time (offset time). Also, in non-preemptive channel scheduling algorithms, once a burst is scheduled on the output port, it is guaranteed to be transmitted without being further segmented. The advantage of the preemptive approach can be observed while incorporating QoS into channel scheduling. In this case, a higher priority unscheduled burst can preempt an already scheduled lower priority data burst.

In order to implement a non-preemptive scheme, we need to use head dropping on the unscheduled burst for non-void-filling-based scheduling algorithms. We also need the ability to drop both the head and tail of an unscheduled burst for void-filling-based scheduling algorithms. In order to implement preemptive schemes, we need to use tail dropping on the scheduled burst for non-void-filling-based scheduling algorithms, and we may need to drop both the head and the tail of overlapping scheduled bursts for void-filling-based scheduling algorithms. In the void filling case, if the unscheduled burst overlaps more than two bursts, then we resolve one contention at a time.

In order to handle contentions during channel scheduling, several existing algorithms have been modified to work in conjunction with fiber delay lines (FDLs). For example, if the overlap of contention on one of

[1]Bursts which have been assigned a data channel are referred as the *scheduled bursts*, and the burst which arrives to the node waiting to be scheduled as the *unscheduled burst.*

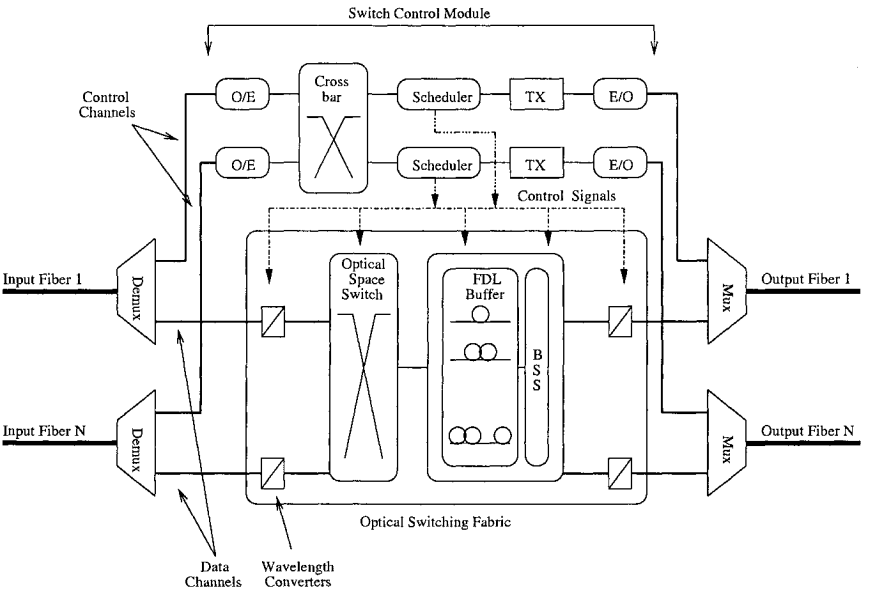


Figure 6.3. Block diagram of an OBS core node.

the data channels is minimal, FDLs may be used to shift the burst by the duration of the overlap, and hence the burst may be successfully scheduled on an outgoing data channel. In [13], the LAUC and LAUC-VF scheduling algorithms have been discussed in conjunction with FDLs. The authors also talk about the dimensioning of FDL buffers. Although the use of FDLs in scheduling reduces the packet loss probability, FDLs introduce a per-hop delay that can affect the end-to-end delay of the data transmitted.

In the rest of this chapter, we study segmentation-based non-preemptive scheduling algorithms with and without FDLs for OBS networks. We compare these non-preemptive scheduling algorithms with existing scheduling algorithms in terms of packet loss performance.

6.2 OBS Core Node Architecture

Figure 6.3 shows a typical architecture of an optical burst-switched node, where optical data bursts are received and sent to the neighboring nodes through physical fiber links. The architecture consists primarily of wavelength converters, variable FDLs, an optical space switch, and a switch control module. We assume that all the header packets incur a fixed processing time at every intermediate node. The switch control module processes the BHPs and sends the control information to the

switching fabric to configure the wavelength converters, space switch, and broadcast and select switch for the associated data burst. It is important to note that the arrangement of the key components depends on the architecture of OBS node considered. A number of different OBS node architectures are possible using FDLs as optical buffers.

Two OBS node architectures with FDLs are considered for realizing the segmentation-based scheduling algorithms. The architecture in Fig. 6.4(a) shows an input-buffered OBS node with FDLs dedicated to each input port, while Fig. 6.4(b) shows an output-buffered OBS node with FDLs dedicated to each output port.

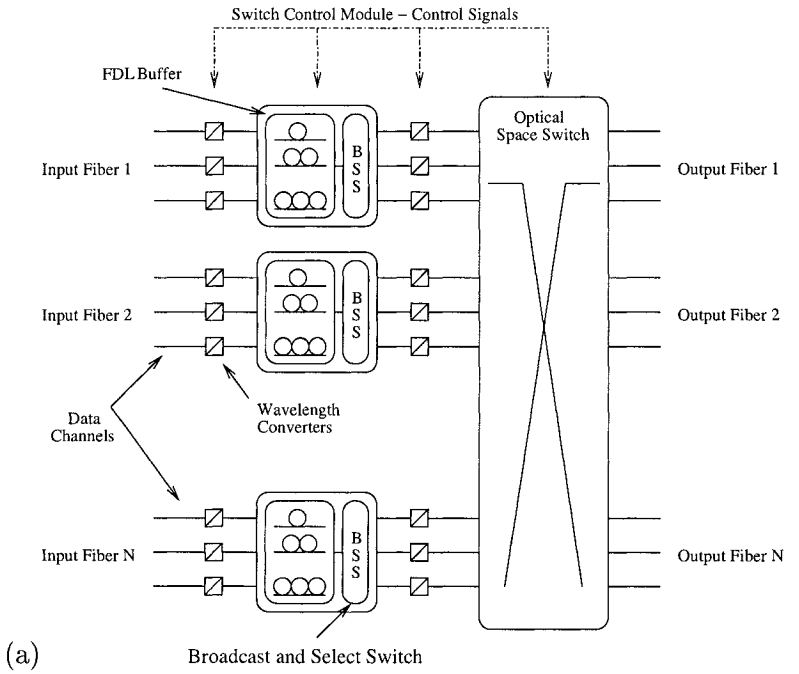
In the input-buffered OBS node architecture shown in Fig. 6.4(a), each input port is equipped with an FDL buffer containing N delay lines. The input-buffered architecture supports the *delay-first* scheduling algorithms. The n data channels are demultiplexed from each input fiber link and are passed through wavelength converters whose function is to convert the input wavelengths to wavelengths that are used within the FDL buffers. The use of different wavelengths in the FDL buffers and on the output links helps to resolve contentions among multiple incoming data bursts competing for the same FDL and the same output link. In the design of FDL buffers, we can have fixed delay FDL buffers, variable delay FDL buffers, or a mixture of both. In this work, we follow the architecture with variable delay FDL buffers.

In the output-buffered OBS node architecture, shown in Fig. 6.4(b), the FDL buffers are placed after the switch fabric. The output-buffered architecture supports the *segment-first* scheduling algorithms. The input wavelength converters are used to convert the input wavelengths to the wavelengths that are used within the switching fabric. The functions of the output wavelength converters are the same as described in the input-buffer FDL architecture.

In this chapter, we only consider the above described *per-port* FDL architecture. In order to minimize switch cost, a *per-node* FDL architecture can be adopted, in which a single set of FDLs can be used for all the ports in a node. This lowering of switch cost results in lower performance with respect to packet loss due to increased contention for FDLs.

6.3 Segmentation-Based Non-Preemptive Scheduling Algorithms

The algorithm may need to maintain several channel information such as, the latest available unscheduled time (LAUT) or the horizon, the gaps, and the voids on every outgoing data channel. The following information is used by the scheduler for all the scheduling algorithms:



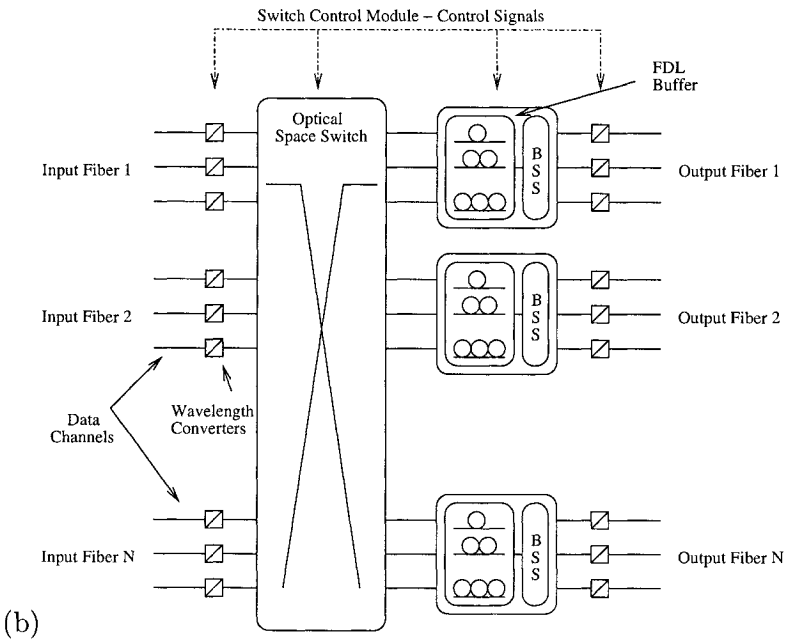


Figure 6.4. (a) Input-buffer FDL Architecture, and (b) Output-buffer FDL Architecture.

- L_b: Unscheduled burst length duration.
- t_{ub}: Unscheduled burst arrival time.
- W: Maximum number of outgoing data channels.
- N_b: Maximum number of data bursts scheduled on a data channel.
- D_i: i^{th} outgoing data channel.
- $LAUT_i$: LAUT of the i^{th} data channel, $i = 1, 2, ..., W$, for non-void filling scheduling algorithms.
- $S_{(i,j)}$ and $E_{(i,j)}$: Starting and ending times of each scheduled burst, j, on every data channel, i, for void filling scheduling algorithms.
- Gap_i: If the channel is available, gap is the difference between t_{ub} and $LAUT_i$ for scheduling algorithms without void filling, and is the difference between t_{ub} and $E_{(i,j)}$ of previous scheduled burst, j, for scheduling algorithms with void filling.

  If the channel is busy, Gap_i is set to 0. Gap information is useful to select a channel for the case in which more than one channel is free.
- $Overlap_i$: Duration of overlap between the unscheduled burst and scheduled burst(s). Overlap is used in non-void filling channel scheduling algorithms. The overlap is zero if the channel is available, otherwise the overlap is the difference between $LAUT_i$ and t_{ub}.
- $Loss_i$: Number of packets dropped due to the assignment of the unscheduled burst on i^{th} data channel. The primary goal of all scheduling algorithms is to minimize loss; hence, loss is the primary factor for choosing a data channel. In case the loss on more than one channel is the same, then other channel parameters are used to reach a decision on the selection of data channel.
- $Void_{(i,k)}$: Duration of k^{th} void on i^{th} data channel. This information is relevant to void filling algorithms. A void is the duration between the $S_{(i,j+1)}$ and $E_{(i,j)}$ on a data channel. Void information is useful in selecting a data channel in case more than one channel is free.

6.3.1 Non-preemptive Minimum Overlap Channel (NP-MOC):

NP-MOC algorithm is an improvement of the existing LAUC scheduling algorithm. The NP-MOC scheduling algorithm keeps track of the LAUT on every data channel. For a given unscheduled burst, the

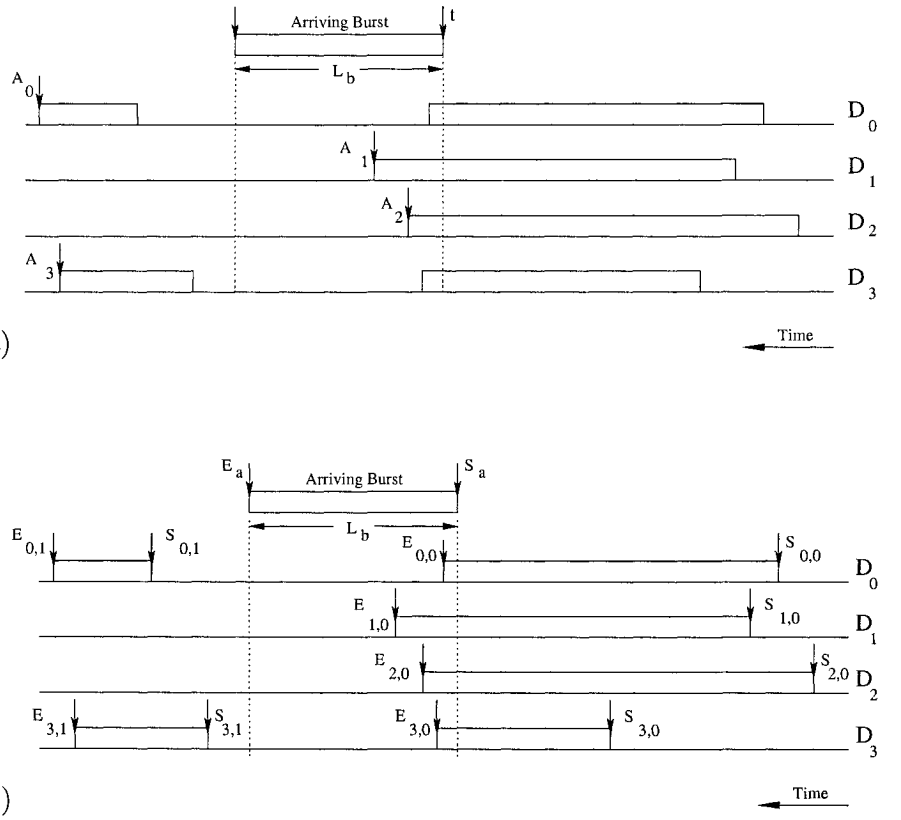


Figure 6.5. Initial data channel assignment using a) non-void filling and b) void filling scheduling.

scheduling algorithm considers all outgoing data channels and calculates the overlap on every channel and chooses the data channel with minimum overlap. case we find more

For example, applying the NP-MOC algorithm to the example in Fig. 6.5(a), we see that data channel D_2 has the minimum loss, and the unscheduled burst is scheduled on D_2 (Fig. 6.6(a)). Here, only the overlapping segments of the unscheduled burst are dropped instead of the entire unscheduled burst as in the case of LAUC. The time complexity of the NP-MOC algorithm is $O(\log W)$.

6.3.2 Non-preemptive Minimum Overlap Channel with Void Filling (NP-MOC-VF):

The NP-MOC-VF scheduling algorithm maintains starting and ending times of each data burst on every data channel. The goal is to

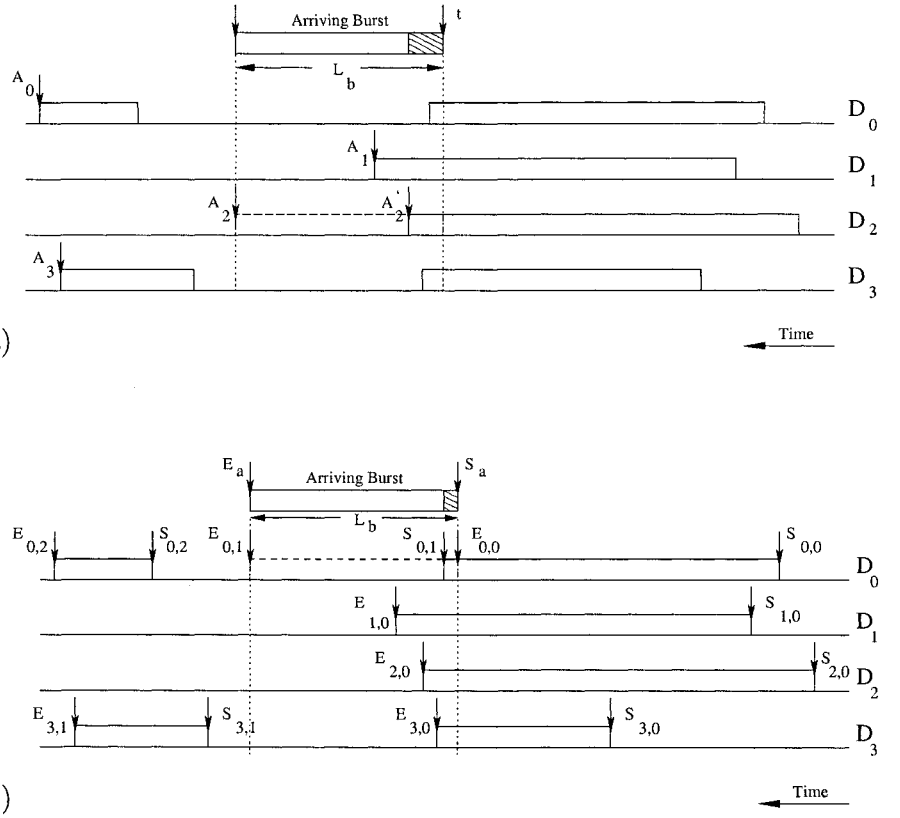


Figure 6.6. Illustration of non-preemptive (a) NP-MOC scheduling algorithm, and (b) NP-MOC-VF scheduling algorithm.

utilize voids between data burst assignments on every data channel. The data channel with a void that minimizes the Gap_i is chosen in case of more than one available channel. If no channel is free, the channel with minimum loss is assigned to the unscheduled burst. are given below. For example, applying the NP-MOC-VF algorithm to the example in Fig. 6.5(b), we see that data channel D_0 has the minimum overlap, and the unscheduled burst is scheduled on D_0 (Fig. 6.6(b)). Here, only the overlapping segments of the unscheduled burst are dropped instead of the entire unscheduled burst as in the case of LAUC-VF. The time complexity of the NP-MOC-VF algorithm is $O(\log(WN_b))$.

Table 6.1 compares all the traditional and proposed channel scheduling algorithms in terms of time complexity and the amount of state information stored. We observe that the time complexity of the non-void filling algorithms is less than that of the void filling algorithms. Also, void filling algorithms, such as LAUC-VF and NP-MOC-VF, store

Table 6.1. Comparison of Segmentation-based Non-preemptive Scheduling Algorithms

Algorithm	Time Complexity	State Information
LAUC	$O(\log W)$	$LAUT_i$, Gap_i
LAUC-VF	$O(\log(WN_b))$	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i
NP-MOC	$O(\log W)$	$LAUT_i$, Gap_i
NP-MOC-VF	$O(\log(WN_b))$	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i

more state information as compared to non-void filling algorithms, such as LAUC and NP-MOC.

6.4 Segmentation-Based Non-Preemptive Scheduling Algorithms with FDLs

There has been substantial work on scheduling using FDLs in OBS [13–1, 8]. In this section, we describe a number of segmentation-based non-preemptive scheduling algorithms incorporating FDLs. Based on the two FDL architectures presented in Section6.2, we have two families of scheduling algorithms. Scheduling algorithms based on the input-buffer FDL node architecture are called *delay-first* scheduling algorithms, while scheduling algorithms based on the output-buffer FDL node architecture are called *segment-first* scheduling algorithms. In both schemes, we assume that full wavelength conversion, FDLs, and segmentation techniques are used to resolve burst contention for an output data channel. However, the order of applying the above techniques depends on the FDL architecture. In delay-first schemes, we resolve contention by FDLs, wavelength conversion, and segmentation, in that order, while in segment-first schemes, we resolve contention by wavelength conversion, segmentation, and FDLs, in that order. Before going on to the detailed description of the schemes, it is necessary to discuss the motivation for developing two different schemes. In delay-first schemes, FDLs are primarily used to delay the entire burst, while in segment-first schemes, FDLs are primarily used to delay the segmented bursts. Delaying the entire burst and then segmenting the burst keeps the packets in order; however, when delaying segmented bursts, packet order is not always maintained. of FDLs. If In general, segment-first schemes will incur lower delays than delay-first schemes. In both the schemes, the sched-

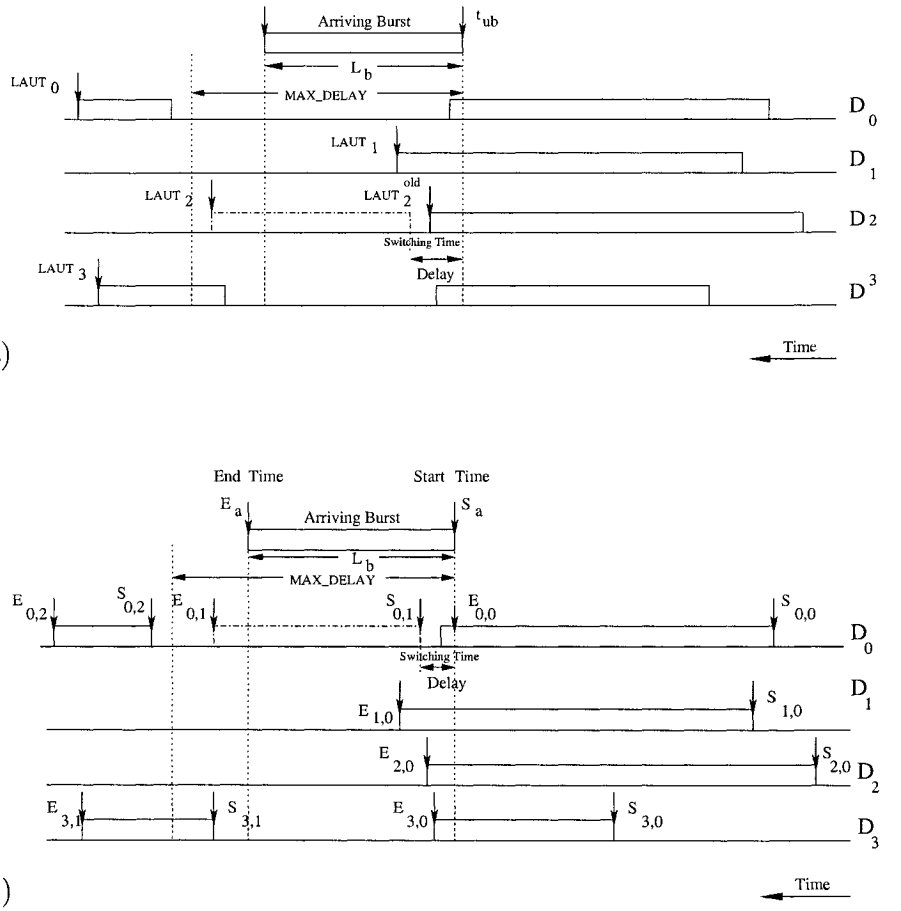


Figure 6.7. Illustration of (a) NP-DFMOC algorithm, and (b) NP-DFMOC-VF algorithm.

uler has to additionally know MAX_DELAY, i.e., the maximum delay provided by the FDLs.

We will now describe the segmentation-based non-preemptive scheduling algorithms which use segmentation, wavelength conversion, and FDLs.

6.4.1 Delay-First Scheduling Algorithms

Non-preemptive Delay-First Minimum Overlap Channel (NP-DFMOC):

The NP-DFMOC algorithm calculates the overlap on every channel and then selects the channel with minimum overlap. If a channel is available, then the unscheduled burst is scheduled on the free channel with

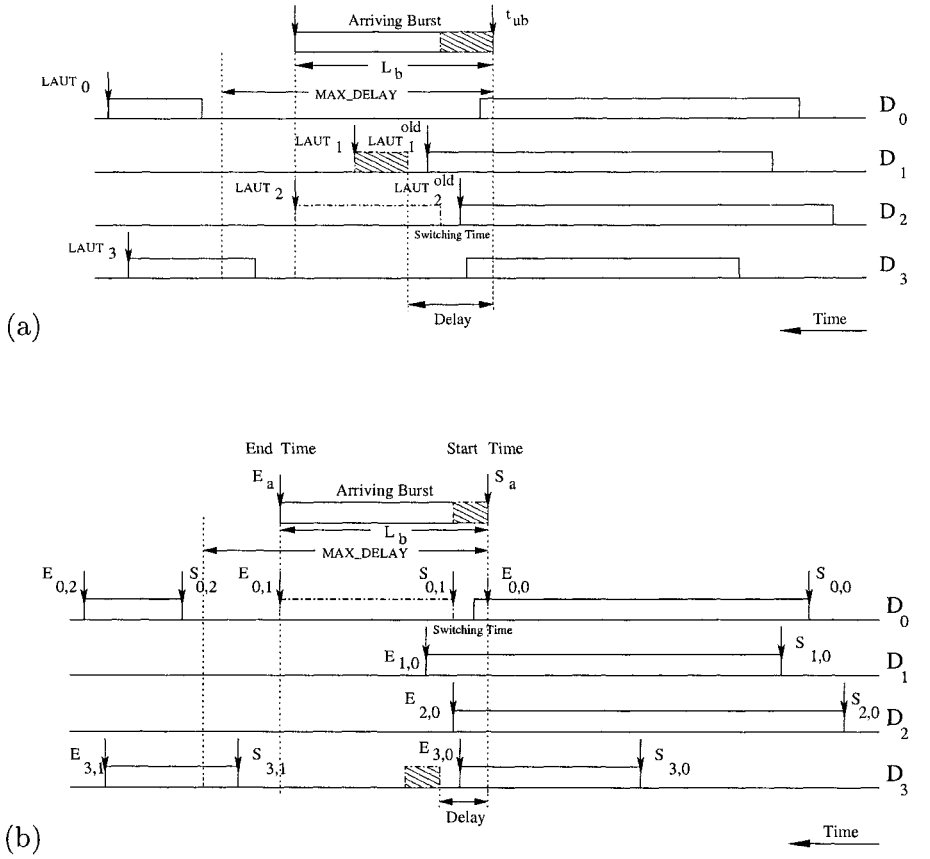


Figure 6.8. Illustration of (a) NP-SFMOC algorithm, and (b) NP-SFMOC-VF algorithm.

the minimum gap. If all channels are busy and the minimum overlap is greater than or equal to the sum of the unscheduled burst length and MAX_DELAY, then the entire unscheduled burst is dropped. Otherwise, the unscheduled burst is delayed for the duration of the minimum overlap and scheduled on the selected channel. In case the minimum overlap is greater than MAX_DELAY, the unscheduled burst is delayed for MAX_DELAY and the non-overlapping burst segments of the unscheduled burst is scheduled, while the overlapping burst segments are dropped. For example, in Fig. 6.7(a), the data channel D_2 has the minimum overlap, thus the unscheduled burst is scheduled on D_2 after providing a delay using FDLs.

Non-Preemptive Delay-First Minimum Overlap Channel with Void Filling (NP-DFMOC-VF):

The NP-DFMOC-VF algorithm calculates the delay until the first void on every channel and then selects the channel with minimum delay. If a channel is available, the unscheduled burst is scheduled on the free channel with minimum gap. If all channels are busy and the starting time of the first void is greater than or equal to the sum of the end time, E_a, of the unscheduled burst and MAX_DELAY, then the entire unscheduled burst is dropped. length and MAX_DELAY, Otherwise, the unscheduled burst is delayed until the start of the first void on the selected channel, where the non-overlapping burst segments of the unscheduled burst are scheduled, while the overlapping burst segments are dropped. In case the start of the first void is greater than the sum of the start time, S_a, of the unscheduled burst and MAX_DELAY, then the unscheduled burst is delayed for MAX_DELAY and the non-overlapping burst segments of the unscheduled burst are scheduled, while the overlapping burst segments are dropped. For example, consider Fig. 6.7(b). By applying the NP-DFMOC-VF algorithm, the data channel D_0 has the minimum delay, thus the unscheduled burst is scheduled on D_0 after delaying the burst using FDLs. In this case, only the overlapping segments of the burst are dropped instead of the entire burst as in the case of LAUC-VF.

6.4.2 Segment-First Scheduling Algorithms

Non-preemptive Segment-First Minimum Overlap Channel (NP-SFMOC):

The NP-SFMOC algorithm calculates the overlap on every channel and then selects the data channel with minimum overlap. If a channel is available, the unscheduled burst is scheduled on the free channel with the minimum Gap_i. If all channels are busy and the minimum overlap is greater than or equal to the sum of the unscheduled burst length and MAX_DELAY, then the entire unscheduled burst is dropped. Otherwise, the unscheduled burst is segmented (if necessary) and the non-overlapping burst segments are scheduled on the selected channel, while the overlapping burst segments are re-scheduled. Next, the algorithm calculates the overlap on all the channels for the re-scheduled burst segments. The re-scheduled burst segments are delayed for the duration of the minimum overlap and scheduled on the selected channel. In case the minimum overlap is greater than MAX_DELAY, then the re-scheduled burst segments are delayed for MAX_DELAY and the non-overlapping burst segments of the re-scheduled burst segments are scheduled, while the overlapping burst segments are dropped. For example, in Fig. 6.8(a), we observe that the data channel D_2 has the minimum overlap for the

unscheduled burst, thus the unscheduled burst is scheduled on D_2, and the re-scheduled burst segments are scheduled on D_1.

Non-preemptive Segment-First Minimum Overlap Channel with Void Filling (NP-SFMOC-VF):

The NP-SFMOC-VF algorithm calculates the loss on every channel and then selects the channel with minimum loss. If a channel is available, the unscheduled burst is scheduled on the free channel with minimum gap. If all channels are busy and the starting time of the first void is greater than or equal to the sum of the end time, E_a, of the unscheduled burst and MAX_DELAY, then the entire unscheduled burst is dropped. If the starting time of the first void is greater than or equal to the end time, E_a, of the unscheduled burst, the NP-DFMOC-VF algorithm is employed.

Otherwise, the unscheduled burst is segmented (if necessary) and the non-overlapping burst segments are scheduled on the selected channel, while the overlapping burst segments are re-scheduled. For the re-scheduled burst segments, the algorithm calculates the delay required until the start of the next void on every channel and selects the channel with minimum delay. The re-scheduled burst segments are delayed until the start of the first void on the selected channel. The non-overlapping burst segments of the re-scheduled burst are scheduled, while the overlapping burst segments are dropped. In case the start of the next void is greater than the sum of the start time, S_a, of the unscheduled burst and MAX_DELAY, the re-scheduled burst segments are delayed for MAX_DELAY and the non-overlapping burst segments of the re-scheduled burst are scheduled, while the overlapping burst segments are dropped. For example, in Fig. 6.8(b), we observe that the data channel D_0 has the minimum loss, thus the unscheduled burst is scheduled on D_0, and the unscheduled burst segments are scheduled on D_3 (as it incurs the minimum delay) after providing a delay using FDLs.

Table 6.2 compares all of the discussed segmentation-based non-preemptive channel scheduling algorithms with FDLs in terms of time complexity and the amount of state information stored. We can observe that the time complexity of the non-void filling algorithms is less than the void filling algorithms. Also, void filling algorithms, such as, LAUC-VF, NP-DFMOC-VF, and NP-SFMOC-VF, store more state information as compared to non-void filling algorithms, such as LAUC, NP-DFMOC, and NP-SFMOC.

6.5 Numerical Results

In order to evaluate the performance of the proposed channel scheduling algorithms, a simulation model is developed. Burst arrivals to the

Table 6.2. Comparison of Segmentation-based Non-preemptive Scheduling Algorithms with FDLs

Algorithm	Time Complexity	State Information
LAUC	$O(\log W)$	$LAUT_i$, Gap_i
LAUC-VF	$O(\log(WN_b))$	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i
NP-DFMOC	$O(\log W)$	$LAUT_i$, Gap_i
NP-DFMOC-VF	$O(\log(WN_b))$	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i
NP-SFMOC	$O(\log W)$	$LAUT_i$, Gap_i
NP-SFMOC-VF	$O(\log(WN_b))$	$S_{(i,j)}$, $E_{(i,j)}$, Gap_i

network are Poisson, andeach burst length is an exponentially generated random number rounded to the nearest integer multiple of the fixed-sized packet length of 1250 bytes. The average burst length is 100 μs. The link transmission rate is 10 Gb/s. Current switching technologies provide us with a range of switching times from a few ms (MEMS) [15] to a few ns (SOA-based) [16]. We assume a conservative switch reconfiguration time of 10 μs. The burst header processing time at each node depends on the architecture of the scheduler and the complexity of the scheduling algorithm. Based on current CPU clock speeds and a conservative estimate of the number of instructions required, we assume burst header processing time to be 2.5 μs. We know that in any optical buffer architecture, the size of the buffers is severely limited, not only by signal quality concerns, but also by physical space limitations. To delay a single burst for 1 ms requires over 200 km of fiber. Due to this size limitation of optical buffers, we consider a maximum FDL delay of 0.01 ms. Traffic is uniformly distributed over all sender-receiver pairs. Fixed minimum-hop routing is used to find the path between all node pairs. All the simulation are implemented on the standard 14-node NSF network shown in Fig. 6.9, where link distances are in km.

Figure 6.10(a) plots the total packet loss probability versus load for different channel scheduling algorithms, with 8 data channels on each link. We observe that the segmentation-based channel scheduling algorithms perform significantly better than algorithms without segmentation. The proposed segmentation-based scheduling algorithms perform better than the algorithms without segmentation because, when contention occurs, only the overlapping packets from one of the bursts are

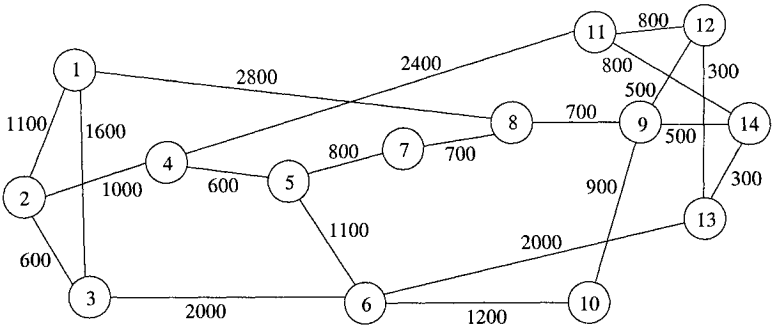


Figure 6.9. 14-Node NSF Network.

lost instead of the entire burst. We see that NP-MOC suffers lower loss as compared to LAUC. Also, NP-MOC-VF performs better than LAUC-VF. We can also observe that NP-MOC and NP-MOC-VF are the best algorithms without and with void filling respectively. Also, the algorithms with void filling perform better than algorithms without void filling as expected. Note that the plots are in log scale. At a total network input load of 5 Erlang, NP-MOC performs 70% better that LAUC and NP-MOC-VF performs 63% better than LAUC-VF.

Figure 6.10(b) plots the average end-to-end delay versus load for different channel scheduling algorithms, with 8 data channels on each link. We observe that the segmentation-based channel scheduling algorithms have higher average end-to-end packet delay than existing channel scheduling algorithms without segmentation. The higher delay for scheduling algorithms with segmentation is due to the higher probability of a successful transmission between source-destination pairs which are farther apart, while in traditional scheduling algorithms the entire burst is dropped in case of a contention; hence, source-destination pairs close to each other have a higher probability of making a successful transmission, which results in lower average end-to-end packet delay. We see that the NP-MOC algorithm has higher delay than the LAUC algorithm. Also, the NP-MOC-VF algorithm has higher delay than the LAUC-VF algorithm. We can also observe that LAUC has the least average end-to-end packet delay among all the algorithms.

Figure 6.11(a) plots the total packet loss probability versus load for different channel scheduling algorithms with FDLs. We observe that the channel scheduling algorithms with burst segmentation perform better than algorithms without burst segmentation at most loads. Also, the delay-first algorithms have lower loss as compared to the segment-first algorithms. This behavior is due to the possible blocking of the

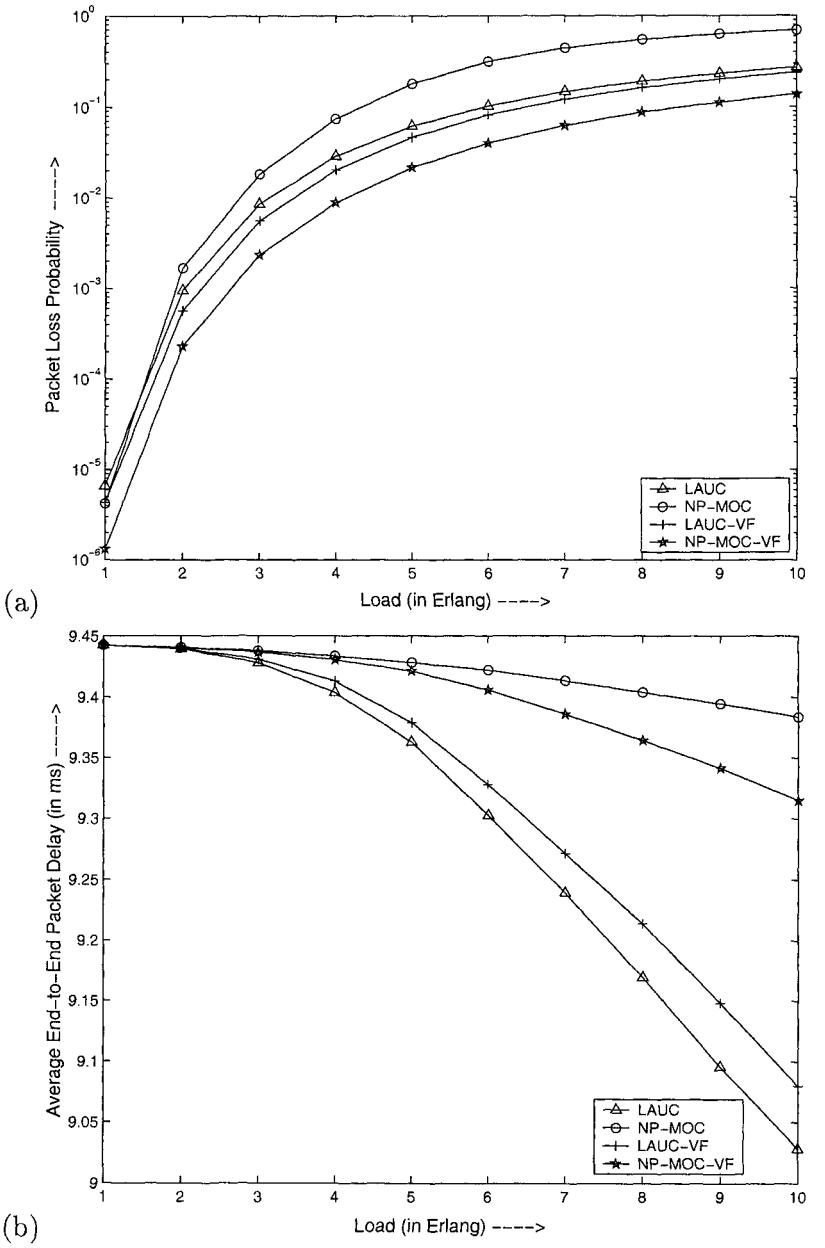


Figure 6.10. (a) Packet loss probability versus load, and (b) average end-to-end delay versus load for different scheduling algorithms with 8 data channels on each link, for the NSF network.

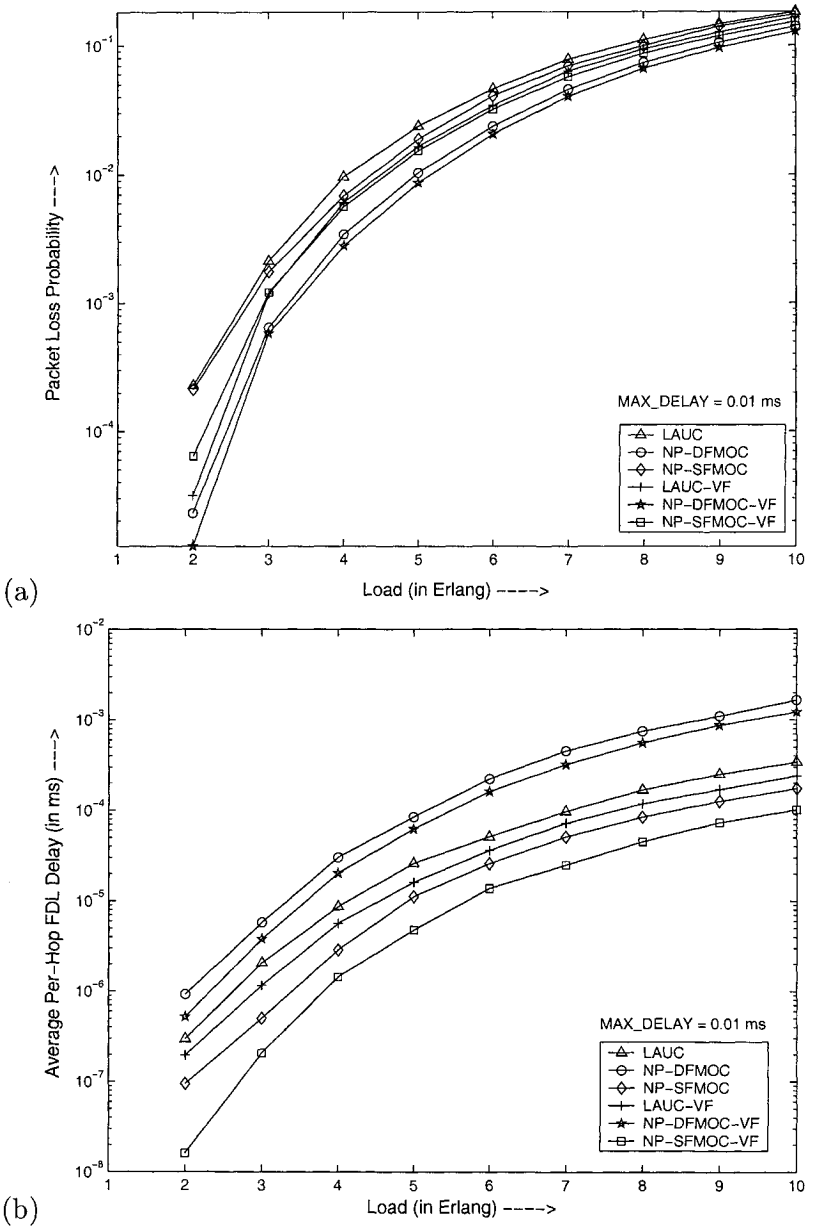


Figure 6.11. (a) Packet loss probability versus load, and (b) average per-hop FDL delay versus load for different scheduling algorithms with 8 data channels on each links and FDLs, for the NSF network.

re-scheduled burst segment by the recently scheduled non-overlapping burst segment in the segment-first algorithms. The loss obtained by delay-first algorithms is the lower bound on delay for the segment-first algorithms. We observe that at any given load, the NP-DFMOC and NP-DFMOC-VF algorithms perform the best, since the unscheduled burst is delayed first; and in the case where there is still a contention, the burst is segmented and only the overlapping burst segment is dropped. The segment-first algorithms lose a number of packets proportional to the switching time every time there is a contention, while the LAUC and LAUC-VF algorithms delay the burst in case of a contention and schedule the burst if the channel is free after the provided delay. Hence, at low loads, LAUC-VF performs better than NP-SFMOC-VF, and, as the load increases, NP-SFMOC-VF performs better. Therefore a substantial gain is achieved by using segmentation and FDLs.

Figure 6.11(b) plots the average per-hop FDL delay versus load for different channel scheduling algorithms. We observe that the delay-first algorithms have higher per-hop FDL delay as compared to the segment-first algorithms, since FDLs are the primary contention resolution technique in the delay-first algorithms, and segmentation is the primary contention resolution technique in the segment-first algorithms. We also observe that the per-hop FDL delay of void filling algorithms is lower than the delay for non-void filling algorithms, since the scheduler can of assign the arriving bursts to closer voids that incur lower FDL delay as compared to scheduling the bursts at the end of the horizon (LAUT) in the case of non-void filling algorithms. contention; Hence, we can carefully choose either delay-first or segment-first schemes based on loss and delay tolerances of input IP packets.

When a high MAX_DELAY value is used, algorithms which use FDLs as the primary contention resolution technique, such as LAUC, LAUC-VF, NP-DFMOC, NP-DFMOC-VF, outperform the algorithms which use segmentation as the primary contention resolution technique, such as NP-SFMOC, NP-SFMOC-VF [10].

In this chapter, we considered burst segmentation and FDLs with wavelength conversion for burst scheduling in optical burst-switched networks, and we discussed a number of data channel scheduling algorithms for optical burst-switched networks. The segmentation-based scheduling algorithms perform better than the existing scheduling algorithms with and without void filling in terms of packet loss. We also introduced two categories of scheduling algorithms based on the FDL architecture. The delay-first algorithms are suitable for transmitting packets which have higher delay tolerance and strict loss constraints, segment-first algorithms are suitable for transmitting packets which have higher loss

tolerance and strict delay constraints. An interesting area of future work would be to implement the preemptive scheduling algorithms for providing QoS support in the optical burst-switched networks.

References

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

[2] L. Tancevski, A. Ge, G. Castanon, and L. Tamil. A new scheduling algorithm for asynchronous, variable length IP traffic incorporating void filling. In *Proceedings, Optical Fiber Communication Conference (OFC)*, volume 3, pages 180–182, February 1999.

[3] M. Iizuka, M. Sakuta, Y. Nishino, and I. Sasase. A scheduling algorithm minimizing voids generated by arriving bursts in optical burst switched WDM network. In *Proceedings, IEEE Globecom*, volume 3, pages 2736–2740, 2002.

[4] J. Xu, C. Qiao, J. Li, and G. Xu. Efficient channel scheduling algorithms in optical burst switched networks. In *Proceedings, IEEE Infocom*, volume 3, pages 2268–2278, March 2003.

[5] F. Farahmand and J. P. Jue. Look-ahead window contention resolution in optical burst switched networks. In *Proceedings, IEEE Workshop on High Performance Switching and Routing*, June 2003.

[6] S. Charcranoon, T. S. El-Bawab, H. C. Cankaya, and J. Shin. Group-scheduling for optical burst switched (OBS) networks. In *Proceedings, IEEE Globecom*, pages 2745–2749, December 2003.

[7] Y. Xiong, M. Vanderhoute, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1838–1854, October 2000.

[8] C. Gauger. Dimensioning of FDL buffers for optical burst switching nodes. In *Proceedings, IFIP Conference on Optical Network Design and Modeling (ONDM)*, February 2002.

[9] V. M. Vokkarane and J. P. Jue. Burst segmentation: An approach for reducing packet loss in optical burst switched networks. *SPIE Optical Networks Magazine*, 4(6):81–89, November-December 2003.

[10] V.M. Vokkarane, G.P.V. Thodime, V.B.T. Challagulla, and J.P. Jue. Channel scheduling algorithms using burst segmentation and FDLs for optical burst-switched networks. In *Proceedings, IEEE International Conference on Communications (ICC)*, volume 2, pages 1443–1447, May 2003.

[11] V. M. Vokkarane and J. P. Jue. Segmentation-based non-preemptive scheduling algorithms for optical burst-switched networks. In *Proceedings, First International Workshop on Optical Burst Switching (WOBS), co-located with OptiComm 2003*, October 2003.

[12] C. Gauger. Performance of converter pools for contention resolution in optical burst switching. In *Proceedings, SPIE OptiComm*, July 2002.

[13] C. Gauger. Contention resolution in optical burst switching networks. In *Advanced Infrastructures for Photonic Networks: WG 2 Intermediate Report*, pages 62–82, 2002.

[14] J. Ramamirtham and J. Turner. Design of wavelength converting switches for optical burst switching. In *Proceedings, IEEE Infocom*, pages 2008–2018, June 2002.

[15] A. Neukermans and R. Ramaswami. Mems technology for optical networking applications. *IEEE Communications Magazine*, 39(1):62–69, January 2001.

[16] L. Rau, S. Rangarajan, D.J. Blumenthal, H.-F.Chou, Y.-J. Chiu, and J.E. Bowers. Two-hop all-optical label swapping with variable length 80 Gb/s packets and 10 Gb/s labels using nonlinear fiber wavelength converters, unicast/multicast output and a single EAM for 80- to 10 Gb/s packet demultiplexing. In *Proceedings, Optical Fiber Communication Conference (OFC)*, pages FD2–1–FD2–3, March 2002.

Chapter 7

QUALITY OF SERVICE

A significant issue for next-generation networks is the ability to support a wide range of services for different types of applications. In order to support these services, the network must be able to provide guarantees as well as differentiation with respect to parameters such as loss and delay. This chapter focuses on the problem of providing quality of service (QoS) in OBS networks.

In general, QoS can be provided in OBS networks by introducing differentiation at some point in the network. Typical approaches for differentiation include differentiated offset times, differentiated contention resolution policies, differentiated burst assembly, and differentiated scheduling.

There are two basic models for QoS: *relative QoS* and *absolute QoS*. In the relative QoS model, the performance of each class is not defined quantitatively in absolute terms. Instead, the QoS of one class is defined relative to other classes. For example, a burst of high priority is guaranteed to experience lower loss probability than a burst of lower priority. However, the loss probability of a high-priority traffic still depends on the traffic load of lower-priority traffic; and no upper bound on the loss probability is guaranteed for the high-priority traffic.

The absolute QoS model provides a worst-case QoS guarantee to applications. This kind of hard guarantee is essential to support applications with delay and bandwidth constraints. Moreover, from a service provider's point of view, the absolute QoS model is preferred in order to ensure that each user receives an expected level of performance. Efficient admission control and resource provisioning mechanisms are needed to support the absolute QoS model.

QoS models can also be classified based on the *degree of isolation* between the different traffic classes. In an *isolated model*, the performance of the high-priority traffic is independent of the low-priority traffic. While, in a *non-isolated model*, the performance of the high-priority traffic is dependent on the low-priority traffic. The degree of isolation can be selected ahead of time and can be satisfied using different techniques.

In IP networks, many queuing disciplines have been developed in order to provide QoS differentiation. Priority queuing (PQ) is a relative differentiation scheme that stores the packets into prioritized queues at each hop, and schedules packets onto an output port only if all packet queues of higher priority are empty. Weighted fair queuing [1] computes virtual finishing time for each packet at the head of each session queue, and transmits the packet with the smallest virtual finishing time. Weighted fair queuing can provide absolute QoS differentiation in the sense that it is able to guarantee a predictable amount of bandwidth and a maximum delay bound for a specific session. On the other hand, a proportional QoS differentiation model was proposed in [2] and [3] in order to provide relative QoS differentiation. Using this model, the relative QoS differentiation is refined and quantified in terms of queuing delay and packet loss probability. Further, in [4] a *dynamic class selection* framework is proposed to provide absolute QoS in which the proportional QoS differentiation approach controls the QoS spacing of each class at every hop, and the users dynamically search for an appropriate class to meet their absolute requirements. In [5], the authors give an overview of recent research on the proportional QoS differentiation model for various QoS metrics, and propose buffer management schemes for achieving absolute service bounds in the proportional QoS differentiation approach.

7.1 Relative QoS in OBS Networks

In OBS networks, several schemes have been proposed to support the relative QoS model. Relative QoS can be supported by using differentiated signaling, differentiated contention resolution, differentiated burst assembly, or differentiated scheduling.

7.1.1 Prioritized Signaling

In OBS networks, it is possible to implement differentiated signaling protocols as a method for providing differentiated QoS in the optical core. JET-based signaling can be used to handle bursty data traffic, while connection-oriented signaling methods, such as TAW, can be used to handle constant data-rate traffic. For the connection-oriented signal-

ing, the ingress node can wait for an acknowledgement that the resources have been reserved before the data is actually transmitted. By combining both connection-oriented and connectionless signaling, the optical core will be capable of supporting a wider range of services.

The basic problem is to determine the criteria for choosing a signaling method. The selection may be based on static parameters, such as QoS requirements and hop distance, or the selection may be based on dynamic parameters, such as the current traffic conditions in the network. In the static selection approach, the problem is to assign a specific signaling method to each burst type in a manner which satisfies the QoS requirements of the packets contained in the burst. In the dynamic approach, the selection of the signaling method is performed on-line and is based on dynamic network state information or the dynamic composition of the burst. For example, if the network load is low, a connectionless signaling method will provide lower delays than a connection-oriented signaling method while still maintaining fairly low packet losses. Under high network loads, a connectionless signaling method may result in an unacceptable level of packet loss; thus, a connection-oriented signaling method may be preferred.

7.1.2 Offset-Based QoS

In [6, 7], an additional-based offset JET scheme was proposed for isolating classes of bursts, such that high-priority bursts experience less contention and loss than low-priority bursts. In the offset-based QoS method, an extra offset time is given to higher-priority bursts. By introducing an extra offset time, resources can be reserved further in advance of the burst's arrival, thereby increasing the probability of a successful reservation.

To illustrate the concept of offset-based QoS, consider an example in which there are two classes of bursts, a high priority class and a low priority class. Let t_a^h and t_a^l be the arrival times of control messages for a high-priority burst and a low priority burst respectively. An offset time of T is given to low-priority bursts, and an offset time of $T + t_{add}$ is given to high-priority bursts. The length of the high-priority burst is given by L_h, and the length of the low-priority burst is given by L_l.

If the control message of the high-priority burst arrives before the control message of the low-priority burst ($t_a^h \leq t_a^l$), then the high-priority burst will always be successfully scheduled. On the other hand, if the control message of the high-priority burst arrives after the control message of the low-priority burst ($t_a^h > t_a^l$), then the high-priority burst can still be scheduled if the starting time of the high-priority burst is after the ending time of the low-priority burst, i.e., $t_a^h + T + t_{add} > t_a^l + T + L_l$.

Thus, in order for the high-priority burst to be scheduled, the additional offset time, t_{add} must be larger than $L_l + (t_a^l - t_a^h)$.

One limitation of offset-based QoS schemes is that scheduled low-priority bursts can still result in the loss of arriving high-priority bursts. A metric for measuring the degree of this effect is called *class isolation.* Class isolation specifies the percentage of high-priority bursts that are unaffected by low-priority bursts. If class isolation is equal to 100%, then the high-priority bursts will not incur any loss due to low-priority bursts. It has been shown that, under certain conditions, an additional offset time of 5 times the maximum burst length is required to achieve a class isolation of 99%. This requirement may result in significant additional delays for high-priority bursts if a high degree of class isolation is desired. Thus, the approach may be capable of satisfying loss requirements, but may not be capable of meeting delay requirements. Furthermore, it has been shown that an offset-based scheme can lead to unfairness, with larger low-priority bursts experiencing higher loss than smaller low-priority bursts [8, 9].

7.1.3 Prioritized Contention Resolution

Another approach for providing QoS is to differentiate between bursts during contention resolution. One approach for differentiated contention resolution based on the concepts of burst segmentation and burst deflection is presented in [10]. In this case, bursts are assigned priorities, and contention between bursts is resolved through selective segmentation, deflection, and burst dropping based on these priorities.

The general problem is approached by first defining the possible segmentation and deflection policies which can be applied when a contention occurs. The possible contention scenarios which can take place between bursts of different priorities and lengths are then defined. Finally, the policy to apply for each specific contention scenario is specified.

When two bursts contend with one another, one of the following policies may be applied to resolve the contention:

- *Segment First and Deflect Policy (SFDP):* The contending burst wins the contention. The original burst is segmented, and the tail segments of the original burst may be deflected if an alternate port is available, otherwise the tail segments of the original burst are dropped.

- *Deflect First and Drop Policy (DFDP):* The contending burst is deflected to an alternate port if an alternate port is available. If no alternate port is available, then the contending burst is dropped.

- *Deflect First, Segment and Drop Policy (DFSDP):* The contending burst is deflected to an alternate port if an alternate port is available. If no alternate port is available, then the original burst is segmented and the tail segments of the original burst are dropped, while the contending burst is routed to the original output port.

- *Segment and Drop Policy (SDP):* The contending burst wins the contention. The original burst is segmented and the tail segments of the original burst are dropped.

- *Drop Policy (DP):* The original burst wins the contention. The entire contending burst is dropped.

There are a total of four different possible contention scenarios, which are based on the priorities and lengths of the original and contending bursts. When two bursts contend, the original burst may be of higher priority than the contending burst, the original burst may be of lower priority than the contending burst, or the two bursts may be of equal priority. For the situation in which bursts are of equal priority, the tie can be broken by considering whether the length of the contending burst is longer or shorter than the remaining tail of the original burst. For each of these four contention scenarios, one of the contention resolution policies described above can be specified.

Figure 7.1.3 illustrates the possible contention scenarios. For the situation in which the contending burst is of lower priority than the original burst, the contending burst should be deflected or dropped; thus, DFDP will be applied. On the other hand, if the contending burst is of higher priority, then it should preempt the original burst. In this situation, SFDP will be applied. For the case in which both bursts are of equal priority, we should attempt to minimize the total number of packets which are dropped or deflected; thus, we compare the length of the contending burst to the remaining length (tail) of the original burst. If the contending burst is shorter than the tail of the original burst, then the contending burst should be deflected or dropped; thus, the DFDP policy is applied. If the contending burst is longer than the tail of the original burst, then we have the option of either attempting to segment and deflect the tail of the original burst, or attempting to deflect the contending burst; thus, either DFSDP or SFDP may be applied. Both options are considered, with the scheme in which DFSDP is applied referred to as Scheme 1, and the scheme in which SFDP is applied referred to as Scheme 2. The policies applied in each scenario under each scheme is summarized in Table 1. The terms P_o and P_c refer to the priorities of the original burst and contending burst respectively, and the terms L_o

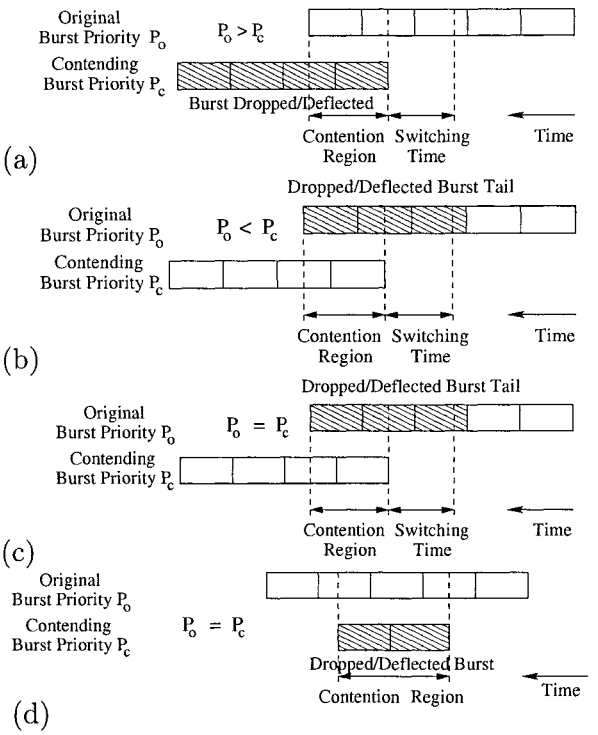


Figure 7.1. (a) Contention of a low-priority burst with a high-priority burst. (b) Contention of a high-priority burst with a low-priority burst. (c) Contention of equal priority bursts with longer contending burst. (d) Contention of equal priority bursts with shorter contending burst.

and L_c refer to the remaining length of the original burst and the length of the contending burst respectively.

Table 7.1. QoS policies for various contention scenarios.

Contention Scenario	Priority Relationship	Length Relationship	Scheme 1 Policies	Scheme 2 Policies
1	$P_o > P_c$	any	DFDP	DFDP
2	$P_o < P_c$	any	SFDP	SFDP
3	$P_o = P_c$	$L_o > L_c$	DFDP	DFDP
4	$P_o = P_c$	$L_o < L_c$	DFSDP	SFDP

Fig. 7.2 shows the performance of the segmentation and deflection schemes for a 14-node nationwide backbone network topology. The plot

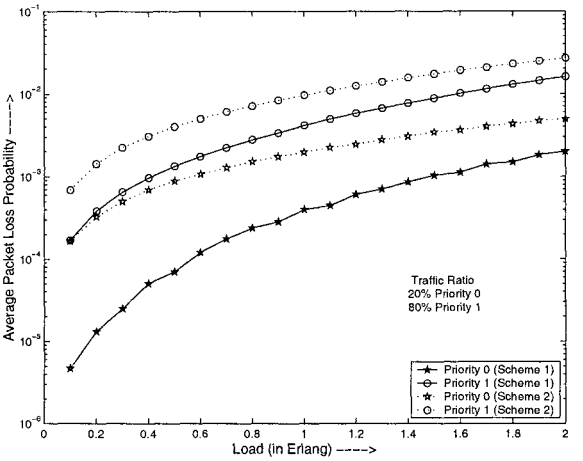


Figure 7.2. Packet loss probability versus load.

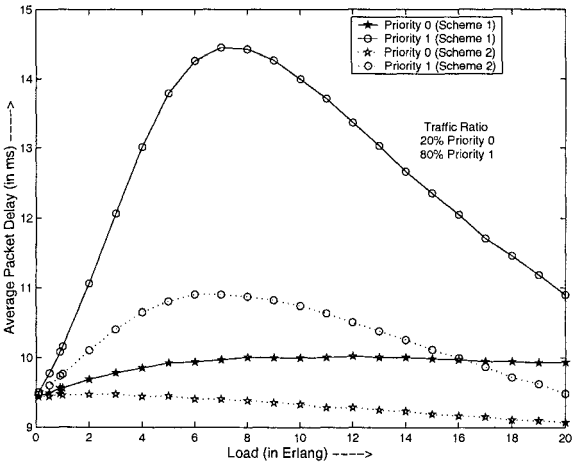


Figure 7.3. Average packet delay versus load.

shows the blocking probability versus load for high-priority (Priority 0) and low-priority (Priority 1) traffic under the assumption that high-priority bursts compose 20% of the overall traffic and low-priority bursts compose 80% of the overall traffic. The results illustrate a significant differentiation between the two burst priorities in terms of packet loss, and also show that, at low loads, the policy of attempting to deflect a contending burst before segmenting a burst performs better than segmenting a burst before attempting to deflect the burst. This behavior is, in part, due to the additional packets lost in the segmentation scheme caused by the switching time. However, at higher loads (not shown),

it was found that Scheme 2 outperforms Scheme 1. The higher packet loss for Scheme 1 under high loads is due to the additional load on the network caused by deflections.

Figure 7.3 shows the difference in delay between the two burst priorities. It can be observed that, even though Scheme 1 provides lower packet loss than Scheme 2, Scheme 1 results in higher delays due to the additional deflections. The delay decreases at higher loads due to the fact that, at higher loads, bursts which travel fewer hops are more likely to successfully reach their destination than bursts which travel a greater number of hops.

7.1.4 Proportional QoS with Early Dropping

In [8], an approach is introduced in which low-priority bursts are intentionally dropped under certain conditions in order to reduce loss for high-priority bursts. The scheme provides a proportional reduction rather than a complete elimination of high-priority burst losses due to contention with low-priority bursts. This proportional QoS scheme based on per-hop information was proposed to support burst loss probability and delay differentiation. The proportional QoS model quantitatively adjusts the QoS metric to be proportional to the differentiation factor of each class. If p_i is the loss metric and s_i is the differentiation factor for Class i, then using the proportional differentiation model, the following will hold for every class,

$$\frac{p_i}{p_j} = \frac{s_i}{s_j}. \tag{7.1}$$

In order to implement this model, each core node needs to maintain traffic statistics, such as the number of burst arrivals and the number of bursts dropped for each class. Hence, the online loss probability of Class i, p_i, is the ratio of the number of Class i bursts dropped to the number of Class i burst arrivals during a fixed time interval. To maintain the differentiation factor between the classes, an intentional burst dropping scheme is employed. A limitation of the scheme is that it can result in the unnecessary dropping of low-priority bursts.

7.1.5 Prioritized Queueing

In [11], another QoS approach based on priority queueing was proposed for OBS networks. The scheme incorporates the LAUC-VF (Section 6.2) scheduling algorithm at the core nodes. The order of assigning channels to the arriving bursts is based on priority queueing, i.e., the higher priority burst are scheduled before the lower priority bursts. Simulation results are presented for the priority scheduling approach with and without FDLs. The authors conclude that the proposed approach

reduces the loss probability of the higher priority bursts, but also leads to significant increase in the loss probability of lower priority bursts.

7.1.6 Reservation-Based QoS

In [12], proportional QoS differentiation is provided by maintaining the number of wavelengths occupied by each class of burst. Every arriving burst is scheduled based on a usage profile maintained at every node. Arriving bursts that satisfy their usage profiles preempt scheduled bursts that do not satisfy their usage profiles, so as to maintain the preset differentiation ratio.

7.1.7 Burst-Assembly-Based QoS

Service differentiation is also provided by different burst assembly schemes. In [8], the waited-time-priority (WTP) scheduler is extended to assemble fixed-length bursts to guarantee flexible packet delay differentiation. Each burst consists of packets of same class. In order to give a controllable burst loss probability for different service classes, lower priority bursts are intentionally dropped in order to provide additional free time to the higher priority bursts. However, this may cause unnecessary burst loss due to intentional dropping.

In [13], the packets are sorted according to the their classes and destination addresses. Each burst consisting of a packet class has a timeout as well as a threshold. When either timeout or threshold is reached, the burst is created and sent into the network. In the case of low packet arrival rate, the threshold of the burst may not be reached and this may lead to smaller bursts due to timeout. Having smaller bursts in the network increases the number of control headers for a given number of packets, in turn leading to higher electronic header processing cost at each intermediate node, which may overload the control plane.

Larger threshold at low arrival rates with lead to higher assembling delay. This may conflict with the time constraint of the packet class. Hence by having packets of different classes into a single burst assembling delay can be lowered [14, 15]. In [13], the lower bound for the burst size and timeout, to avoid the congestion in the control plane is calculated. By assembling packet of different classes into a burst, we reduce the number of control packets for a given number of data packets. This reduces the header processing effort in the core in turn increasing the maximum transmission rate.

In addressing burst assembly, one may consider both the single-class problem, in which there is only one class of packets with specific QoS requirements, and the multi-class problem in which there are multiple

classes of packets, each with different QoS requirements. For the single-class burst assembly problem, the objective is to select appropriate timer or threshold values to either meet delay constraints or to minimize packet loss in the optical core. In order to formulate the multi-class burst assembly problem, we introduce two concepts, referred to as *differentiated burst assembly* and *composite burst assembly.*

In differentiated burst assembly, different classes of packets are assembled into bursts according to different policies. For example, packets which have strict delay requirements may be aggregated using a timer-based policy, while other packets may be aggregated using a threshold-based policy. The performance of differentiated burst assembly schemes at the edge nodes may be complemented by differentiated contention resolution strategies in the optical core.

In composite burst assembly, packets of different classes may be aggregated into a single burst. The motivation for composite burst assembly rests in the observation that, when burst segmentation with a tail-dropping policy is maintained within the optical core, the packets towards the end of the burst are more likely to be dropped than the packets at the head of a burst. Therefore, the location of a packet within a burst and the method of contention resolution in the core will determine the level of service received by the packet.

The generalized burst assembly problem can be formulated as follows. Let N be the number of input packet *classes*, and let M be the number of burst *priorities* supported in the core network. Note that a packet class does not necessarily have a one-to-one correspondence to a burst priority. A set of *burst types* are defined, which specify the burst assembly policies. Let K be the number of burst types. Each burst type is characterized by the following parameters:

- L_k^{MIN}: minimum length of burst of type k.
- L_k^{MAX}: maximum length of burst of type k.
- R_{jk}^{MIN}: minimum number of packets of class j in a burst of type k.
- R_{jk}^{MAX}: maximum number of packets of class j in a burst of type k.
- $S_k = \{j \mid R_{jk}^{MAX} > 0\}$: the set of packet classes which may be included in a burst of type k. When a burst of type k is created, all packets of class $j \in S_k$ will be included in the burst, subject to the constraints specified by $L_k^{MIN}, L_k^{MAX}, R_{jk}^{MIN}$, and R_{jk}^{MAX}.
- P_k: priority assigned to burst of type k. Note that different burst types may have the same priority in the optical core.

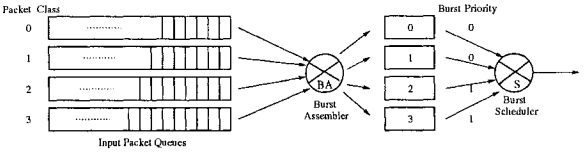


Figure 7.4. Single class per burst.

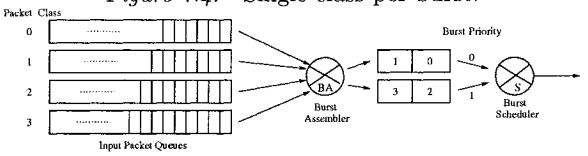


Figure 7.5. Composite burst.

- τ_k: timer value for creating bursts of type k. When a packet of class $j \in S_k$ arrives to a node, a timer is started. When the timer reaches τ_k, a burst of type k is created. If a timer value is not specified for a burst of type k, then the burst is assembled based only on threshold.

- T_k: threshold value for creating bursts of type k. If x_j is the number of packets of class j accumulated at a node, then a burst of type k is created if $\sum_{j \in S_k} x_j \geq T_k$.

Given N, M, and the QoS requirements for each packet class, the burst assembly problem is to choose a value for K, and for each burst type $k = 0, 2, \ldots, K-1$, specify the parameters L_k^{MIN}, L_k^{MAX}, R_{jk}^{MIN}, R_{jk}^{MAX}, P_k, τ_k, and T_k such that the QoS requirements are satisfied.

The important design considerations when defining the burst types are packet loss probability, delay constraints, and bandwidth requirements. Packet loss probability in the optical core is a function of a number of factors, such as burst size, burst priority, number of bursts generated, and contention resolution schemes in the core. Thus, packet loss probability can be affected by adjusting the threshold T_k, the burst size L_k^{MAX}, and the burst priority P_k. End-to-end delay constraints can be met by setting appropriate timer values for each burst type, τ_k. Bandwidth requirements for a given class of packets can be met by ensuring that an adequate number of packets of a given class are inserted into the generated bursts, and that the bursts are generated at a sufficient rate to provide the required bandwidth. Thus, the amount of bandwidth for a given class of packets may be determined by R_{jk}^{MIN} and R_{jk}^{MAX} as well as T_k and τ_k.

Several experiments for a four-class/two-priority network with and without composite bursts were conducted in [14]. Class 0 packets are

assumed to have a delay constraint, while the other packet classes are assumed to have relative packet loss constraints with respect to other packet classes. (i.e., Class 1 packet loss should be less than Class 2 packet loss, etc.). The traffic ratios are assumed to be 10%, 20%, 30%, and 40% for Classes 0, 1, 2, and 3 respectively.

For the case without composite bursts, four burst types can be defined, one for each packet class ($S_k = \{k\}$). All burst types are given the same threshold value, $T_k = 100$ packets, while the burst type corresponding to Class 0 packets is also assigned a timer value of $\tau_0 = 50$ ms. Since there are four burst types, but only two burst priority levels, some burst types must share the same priority. The burst types corresponding to Class 0 and Class 1 packets are both assigned burst Priority 0 ($P_0 = P_1 = 0$), while the burst types corresponding to Class 2 and Class 3 packets are both assigned burst Priority 1 ($P_2 = P_3 = 1$). The burst assembly procedure for this case is shown in Fig. 7.4.

For the case with composite bursts, two burst types can be defined. Burst Type 0 handles Class 0 and Class 1 packets ($S_0 = \{0, 1\}$), while burst Type 1 handles Class 2 and Class 3 packets ($S_1 = \{2, 3\}$). Both burst types are given the same threshold value ($T_0 = T_1 = 100$ packets), while burst Type 0 is assigned a timer value of $\tau_0 = 50$ ms. Burst Type 0 is assigned burst Priority 0 ($P_0 = 0$), and burst Type 1 is assigned burst Priority 1 ($P_1 = 1$). The burst assembly procedure for this case is shown in Fig. 7.5. There are no restrictions on the maximum or minimum number of packets of each class in a given burst, or on the maximum or minimum number of packets in a burst.

Figure 7.6 plots packet loss probability versus load for both the single-class-per-burst case (Single), and the composite-burst case (Composite). A prioritized SDP policy without deflection is utilized in the core. It can be observed that, for the single-class-burst case, there is no difference in packet loss between Class 2 and Class 3 packets, since the burst types for these two classes have the same priority and threshold values. On the other hand, there is a fairly large difference in packet loss between Class 0 and Class 1 packets. Although the burst types for Class 0 and Class 1 have the same priority, the burst type for Class 0 has an additional timer value associated with it. For the composite-burst case, there is a fairly good separation in packet loss among the four classes of traffic.

Figure 7.7 plots the delay versus load. For the single-class-burst case, the delay for Class 0 is fairly constant due to the timer value associated with the burst type. For the other classes of traffic, the delay is proportional to the arrival rate of packets. Since Class 3 traffic arrives at a higher rate than Class 1 or Class 2 traffic, it will hit its threshold sooner, leading to lower delays. For the composite-burst case, both Class 0 and

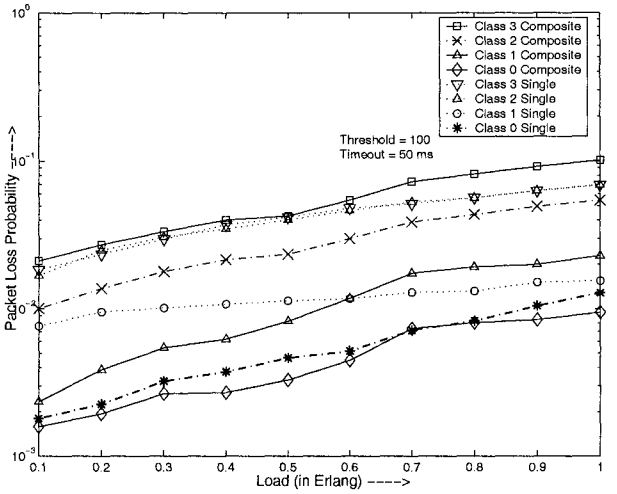


Figure 7.6. Packet loss probability versus load.

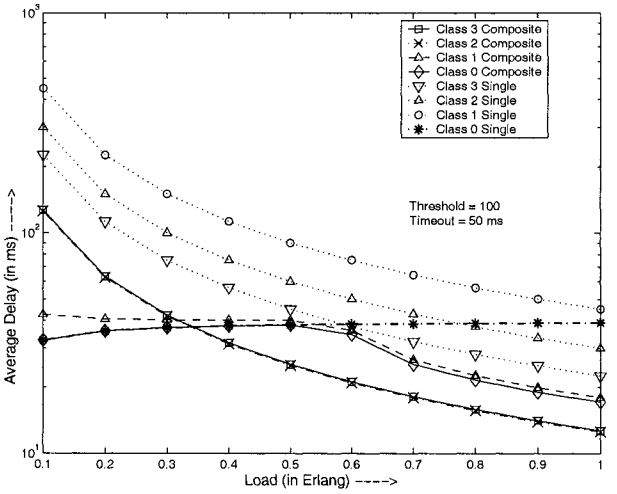


Figure 7.7. Average delay versus load.

Class 1 have fairly constant delay, since both are carried in burst Type 0. Class 2 and Class 3 have matching delays, since they are both carried in burst Type 1. Under higher loads, Class 2 and Class 3 have lower delay than Class 0 and Class 1, because at higher arrival rates, the threshold is reached sooner.

It has been shown that aggregating different classes of packets into a single burst is effective in providing differentiated service with respect to packet loss probability, and that timer-based assembly schemes are

capable satisfying delay requirements. There are a number of additional issues which can be considered in composite burst assembly techniques, such as how many packet classes to include in a given burst type and which packet classes to include in a given burst type. These parameters can be adjusted to address packet loss requirements, delay requirements, and bandwidth requirements of individual packet classes.

When considering how many packet classes to include in a given burst type, there may be a tradeoff between delay and packet loss. If a larger number of packet classes are aggregated into a single burst type, then the threshold is likely to be reached sooner than if there were fewer packet classes associated with the burst type. Since the bursts will be generated more frequently, the delays for these packets will be lower. However, associating a greater number of packet classes to the same burst type is likely to result is a lower degree of differentiation with respect to packet loss. When deciding on which packet classes to include in a given burst type, it may be beneficial to group packet classes which have similar delay requirements. On the other hand, if a tail-dropping burst segmentation contention resolution policy is implemented, then it may be better to place packet classes with high loss tolerance at the tail of every burst type.

Another parameter to investigate is the number of packets of each class to insert into a given burst. In the previous experiments, the number of packets of a given class in a given burst was directly proportional to the arrival rate of that class of packets; however, it may be desirable to place additional restrictions on the number of packets, since the number of packets of a given class in a burst will affect the packet loss probability for that class of packets. If a class of packets occupies only a small fraction of the burst and is located at the head of the burst, there is a high probability that very few of these packets will be dropped. However, if a class of packets occupies a greater fraction of the burst, then that class of packets is likely to experience higher losses, even if the packets are located towards the head of the burst. In order to determine the appropriate number of packets of a given class to insert into a given burst type, the probability of packet loss as a function of a packet's location in a burst of a given type can be determined.

Approaches for satisfying the bandwidth requirements of various traffic classes can also be studied. In order to meet the bandwidth requirements of a given class of packets, we first need to calculate the rate at which bursts of a given type are generated and transmitted. This calculation can be based on timer and threshold values, as well as the burst scheduling policy. Once the burst transmission rate is determined, the bandwidth requirements of a given class of packets can be satis-

fied through the appropriate allocation of capacity within the outgoing bursts.

7.1.8 Look-Ahead Window Contention Resolution

The problem of providing QoS support by implementing a differentiated Look-ahead window Contention Resolution (LCR) algorithm is presented in [16]. In this scheme, bursts are delayed at each node for a certain fixed amount of time. By collecting multiple burst headers over a window of time, more informed decisions can be made as to which bursts to drop. Simulation results show that the look-ahead contention resolution algorithm can readily support service differentiation and offers high overall performance with moderate complexity. A potential disadvantage of this scheme is that bursts will experience additional delay at each node.

7.1.9 Linear Predictive Filter (LPF)-based Forward Resource Reservation

In [17], a Linear Predictive Filter (LPF)-based Forward Resource Reservation method is proposed to reduce the burst delay at edge routers. The authors claim that their QoS strategy achieves burst delay differentiation for different classes of traffic, while maintaining the bandwidth overhead within limits by extending the FRR scheme (aggressive reservation).

7.1.10 QoS in Wavelength-Routed OBS Networks

The authors in [18–21], propose several QoS approaches for WR-OBS networks. In a WR-OBS network, each source node sends a connection request to a centralized request scheduler. At the edge node, the higher-layer traffic is assigned different class of service (CoS) based on the maximum acceptable delay and the destination address. Therefore, each edge node has $C \cdot (N-1)$ buffers, where C is the number of classes and (N-1) is the number of possible destination nodes. At the request scheduler, the connection requests that are sorted based on their class of service into C prioritized request queues. All the higher priority requests are handled before servicing the lower priority request. Since the request scheduler has to handle the connection request of the entire network, the complexity of this approach may be significant.

7.1.11 QoS Based on Physical Signal Quality

In [22], the authors have proposed QoS schemes based on the physical quality of the optical signal, such as signal-to-noise ratio (SNR), maxi-

mum bandwidth, wavelength spacing, and bit error rate (BER). In this scheme, the QoS parameters are specified in the burst header packet and a connection is set up only if all the parameters are satisfied.

7.2 Absolute QoS

Relative QoS differentiation schemes do not provide a worst-case guarantee for any of the supported QoS metrics, thus absolute QoS differentiation schemes are necessary. The most intuitive approach to provide absolute QoS differentiation is to design a hybrid optical backbone network consisting of wavelength-routed lightpaths [23] to carry the guaranteed traffic, and a classical OBS network to carry the non-guaranteed traffic. This approach leads to inefficient usage of bandwidth over the wavelength-routed part of the network. In order to efficiently utilize bandwidth, efficient absolute QoS differentiation schemes need to be developed in which all wavelengths in the network are available for statistical multiplexing and dynamic bandwidth allocation.

7.2.1 Probabilistic Preemptive QoS

In [24], a Probabilistic Preemptive scheme is proposed, for providing service differentiation in terms of burst blocking probability in OBS networks. In this scheme, high-priority class traffic is assigned a preemptive probability. Thus, high-priority bursts can preempt low-priority bursts in a probabilistic manner. The authors claim that by changing the preemptive probability, an OBS node can adjust the ratio of burst blocking probability between different traffic classes, while the overall blocking probability is not affected. The authors in [25] also talk about the concept of introducing a partially preemptive scheduling technique capable of handling data bursts in parts, and may use preemption due to the priorities of data bursts in a multi-service OBS network environment. The Probabilistic Preemptive scheme can also be used to provide absolute QoS in an OBS network.

7.2.2 Early Dropping and Wavelength Grouping

One approach for providing absolute QoS is presented in [26]. In this approach, two different techniques are utilized in order to guarantee that a given class of traffic does not experience loss probability higher than a specified threshold.

The first technique is referred to as *early dropping*, and the second technique is referred to as *wavelength grouping*. In early dropping, bursts that contain packets of lower class traffic may be intentionally dropped, even if there is no contention, in order to support the loss requirements

for the higher-class packets. In the wavelength grouping technique, each class of traffic is assigned a fixed number of wavelength channels to utilize on a given link. The two schemes can either be applied independently of one another, or applied simultaneously in the same network.

In absolute QoS, each service class i is assumed to require a maximum network-wide loss guarantee, $P_{C_i}^{NET}$. Given that each OBS node maintains the same loss guarantee, $P_{C_i}^{MAX}$ for Class i traffic, the $P_{C_i}^{MAX}$ at each node can be calculated from the diameter of the network, D, and $P_{C_i}^{NET}$ as follows,

$$P_{C_i}^{MAX} = 1 - e^{(ln(1-P_{C_i}^{NET}))/D}. \tag{7.2}$$

Therefore, if the per-hop loss probability $P_{C_i}^{MAX}$ is guaranteed at each node along the path, then the network-wide loss probability $P_{C_i}^{NET}$ is guaranteed end-to-end.

Based on the maximum arrival rate of the guaranteed traffic, the routing algorithm, and the network topology, the maximum offered load of the guaranteed traffic on every link can be obtained. For each link, let L_{C_i} be the maximum offered load of Class i traffic, and let W_{C_i} be the minimum number of wavelengths required in order to guarantee that the loss probability of Class i traffic is below $P_{C_i}^{MAX}$. We can compute W_{C_i} for the guaranteed traffic of Class i using the standard Erlang-B formula,

$$\frac{L_{C_i}^{W_{C_i}}/W_{C_i}!}{\sum_{x=0}^{W_{C_i}} L_{C_i}^{x}/x!} \leq P_{C_i}^{MAX}. \tag{7.3}$$

Hence, in order to guarantee the maximum end-to-end loss, each core node must provide at least W_{C_i} wavelengths and must guarantee the maximum per-hop loss probability, $P_{C_i}^{MAX}$, for each Class i traffic.

Early Dropping

In the early dropping mechanism, *an early dropping probability*, $p_{C_i}^{ED}$, is computed for each Class i based on the online measured loss probability and the maximum acceptable loss probability of the immediately-higher-priority class. The Class i burst is indicated to be dropped by an *early dropping flag*, e_i; e_i is set to 1 with a probability of $p_{C_i}^{ED}$, and set to 0 with a probability of $(1-p_{C_i}^{ED})$. In order to decide whether or not to drop the arriving Class i burst, not only does the early dropping flag of Class i need to be considered, but the early dropping flags of all higher priority classes need to be considered. Thus, we generate an *early dropping vector*, ED_i, where $ED_i = \{e_1, e_2, \ldots, e_i\}$ for the arriving Class i burst. The Class i burst is intentionally dropped if $e_1 \vee e_2 \vee \cdots \vee e_i = 1$, that is, the Class i burst is intentionally dropped with a probability

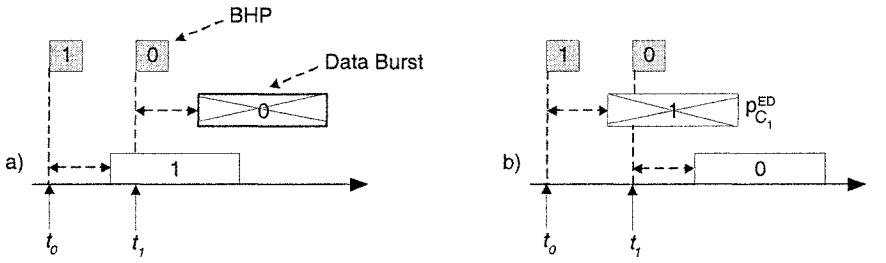


Figure 7.8. (a) Standard Dropping Mechanism, and (b) Early Dropping Mechanism.

of $(1 - \prod_{j=1}^{i}(1 - p_{C_j}^{ED}))$. Note that we do not have an element e_0 for Class 0, since Class 0 has the highest priority.

Consider a two-class example to illustrate the early dropping concept. In Fig. 7.8(a), the BHP of a Class 1 burst (low priority) arrives at time t_0 and reserves the channel. The BHP of a Class 0 burst (high priority) arrives at time t_1, where $t_1 > t_0$, and contends with the Class 1 burst, resulting in the Class 0 burst being dropped. In order to reduce the likelihood of this scenario, a burst of Class 1 is intentionally dropped when $e_1 = 1$, prior to the BHP arrival of the Class 0 burst (Fig. 7.8(b)). e_1 is set to 1 with probability $p_{C_1}^{ED}$; $p_{C_1}^{ED}$ is a function of the maximum acceptable loss probability of Class 0 bursts and the online measured loss probability of Class 0 bursts. The key is to decide when to trigger the early dropping mechanism, and how to compute the early dropping probability.

In order to provide loss guarantees, each OBS core node must monitor the traffic statistics for each guaranteed class. For each output port of an OBS node, let a_{C_i} be the burst arrival counter, and let d_{C_i} be the burst drop counter. We use, $p_{C_i} = (d_{C_i}/a_{C_i})$, as the online measured burst loss probability for the Class i traffic. For this purpose, a_{C_i} and d_{C_i} can be measured within a fixed time window.

We now describe the following *early drop by threshold* and *early drop by span* schemes to compute the early dropping probability, $p_{C_i}^{ED}$, for Class i bursts.

Early Drop by Threshold (EDT)

The basic idea of early drop by threshold (EDT) is to drop the arriving Class i bursts, when the online measured loss probability of Class $(i-1)$, $p_{C_{i-1}}$ reaches the maximum loss probability, $P_{C_{i-1}}^{MAX}$. This early dropping of bursts of the lower-priority classes is a simple way to provide loss guarantee for the higher-priority class. The early dropping probability

of Class i bursts is given by,

$$p_{C_i}^{ED} = \begin{cases} 0 & p_{C_{i-1}} < P_{C_{i-1}}^{MAX} \\ 1 & p_{C_{i-1}} \geq P_{C_{i-1}}^{MAX}, \end{cases} \tag{7.4}$$

where $i \geq 1$.

In the EDT scheme, bursts of each class with lower priority than Class $(i-1)$, suffer from high loss when $p_{C_{i-1}}$ exceeds $P_{C_{i-1}}^{MAX}$. Since there is a single trigger point, the scheme takes extreme steps in order to regulate $p_{C_{i-1}}$.

Early Drop by Span (EDS)

In order to alleviate the side effects of EDT, a scheme known as early drop by span (EDS), that linearly increases $p_{C_i}^{ED}$ as a function of $p_{C_{i-1}}$, is used. Here, a span (range) of acceptable loss probabilities, $\delta_{C_{i-1}}$, for Class $(i-1)$ is chosen. The EDS scheme is triggered when the online measured loss probability of Class $(i-1)$ bursts, $p_{C_{i-1}}$, is higher than $P_{C_{i-1}}^{MIN}$, where, $P_{C_{i-1}}^{MIN} = P_{C_{i-1}}^{MAX} - \delta_{C_{i-1}}$. Thus the early dropping probability of Class i bursts is given by,

$$p_{C_i}^{ED} = \begin{cases} 0 & p_{C_{i-1}} < P_{C_{i-1}}^{MIN} \\ (p_{C_{i-1}} - P_{C_{i-1}}^{MIN})/\delta_{C_{i-1}} & P_{C_{i-1}}^{MIN} \leq p_{C_{i-1}} < P_{C_{i-1}}^{MAX} \\ 1 & p_{C_{i-1}} \geq P_{C_{i-1}}^{MAX}, \end{cases} \tag{7.5}$$

where $i \geq 1$.

The span $(\delta_{C_{i-1}})$ can be chosen as a percentage value of $P_{C_{i-1}}^{MAX}$. We observe that, if $\delta_{C_{i-1}}$ is too high, EDS is triggered prematurely, leading to high loss for lower-priority classes of traffic; while, if $\delta_{C_{i-1}}$ is too low, $p_{C_{i-1}^{ED}}$ will be high, also resulting in high loss for lower-priority classes of traffic.

Wavelength Grouping

This section describes another mechanism, known as *wavelength grouping* for supporting absolute loss guarantee in OBS networks. In the wavelength grouping mechanism, traffic is classified into different groups, and a label is assigned to each group. Each group is provisioned a minimum number of wavelengths. One approach to group the traffic is to assign all traffic of the same service class to the same unique group. Thus, on each link, l, Class i bursts are assigned the same unique local label Li. We obtain W_{C_i} and $P_{C_i}^{MAX}$ for each guaranteed Class i traffic from (7.2) and (7.3). Link l must provide W_{C_i} wavelengths for bursts in the group with assigned Label Li in order to guarantee $P_{C_i}^{MAX}$. If we run out of wavelengths, then the requirement of the remaining guaranteed

class of traffic cannot be satisfied with the given network capacity. On the other hand, if there are certain available wavelengths after provisioning wavelengths for all the guaranteed traffic, the remaining wavelengths are used to carry the non-guaranteed traffic. We propose two schemes for wavelength grouping, namely, *static wavelength grouping* (SWG) and *dynamic wavelength grouping* (DWG).

Static Wavelength Grouping (SWG)

In SWG, a fixed set of wavelengths is dedicated for the traffic within a given group. If W_{C_0} wavelengths on link l are required for bursts in the group with assigned Label $L0$, the first W_{C_0} wavelengths $(w_0, w_1, \ldots, w_{(W_{C_0}-1)})$ are reserved for bursts in this group. Furthermore, bursts labeled $L0$ can only use these $w_0, w_1, \ldots, w_{(W_{C_0}-1)}$ wavelengths on the link l. In the case of guaranteeing more than one class of traffic, the process is repeated until the necessary wavelengths have been reserved for all of the guaranteed traffic. The remaining unreserved wavelengths are used to carry the best-effort traffic. For the scenario shown in Fig. 7.9(a), when a burst labeled $L1$ arrives at time t, it can only be scheduled on Wavelength 3, which is statically preassigned to the bursts labeled $L1$.

Dynamic Wavelength Grouping (DWG)

In DWG, a fixed number of wavelengths, but not necessarily a fixed set of wavelengths, is reserved for the traffic within a given group. To ensure that the number of wavelengths occupied by bursts of a given group does not exceed the number of wavelengths provisioned, the OBS node must keep track of the number of wavelengths currently occupied by bursts of each group. A burst with a given label can be dynamically scheduled onto an available wavelength, if the number of wavelengths currently occupied by bursts of the same label is less than the number of wavelengths provisioned for that group. In Fig. 7.9(b), suppose the number of wavelengths that bursts labeled $L1$ can use is, $W_{C_1} = 1$. When a burst labeled $L1$ arrives at time t, Wavelength 1 and Wavelength 3 are available and no bursts labeled $L1$ are currently scheduled. Hence, the arriving burst is scheduled on Wavelength 1, which is the latest available unscheduled channel.

Comparing SWG and DWG, note that SWG is less complex and simpler to implement. However, DWG has the advantage of being able to dynamically schedule a burst onto the best wavelength based on the channel allocation status of each link, thereby improving network performance.

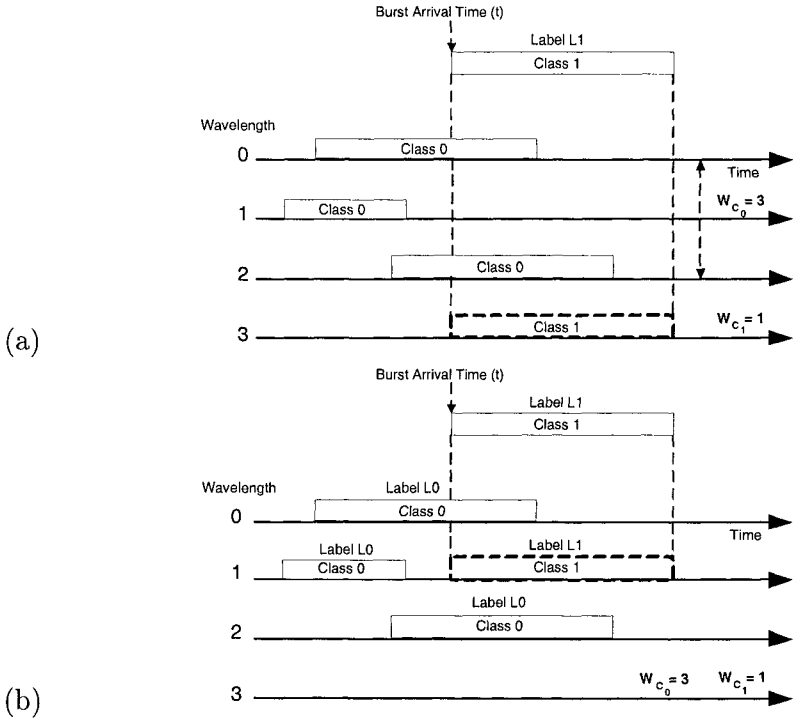


Figure 7.9. Illustration of (a) SWG, and (b) DWG schemes.

7.2.3 Integrated Schemes

Without the help of an early dropping mechanism, the wavelength grouping mechanism schedules the bursts of a given class only on a limited number of wavelengths, even when the loss probabilities of other classes of traffic are much lower than their required maximum loss probability. This restriction results in inefficient wavelength utilization. Therefore, the early dropping mechanism can be integrated with the wavelength grouping mechanism to achieve better performance. In the early dropping mechanism, EDS has significantly better loss performance than EDT, based on simulation results (Fig. 7.10); hence, EDS is integrated with the wavelength grouping schemes. In the integrated schemes, EDS assigns a local label to each burst based on the class of the burst and the current value of the corresponding early dropping vector. The wavelength grouping mechanism provisions a minimum number of wavelengths for each group of traffic with the same label and schedules each burst based on the provisioning.

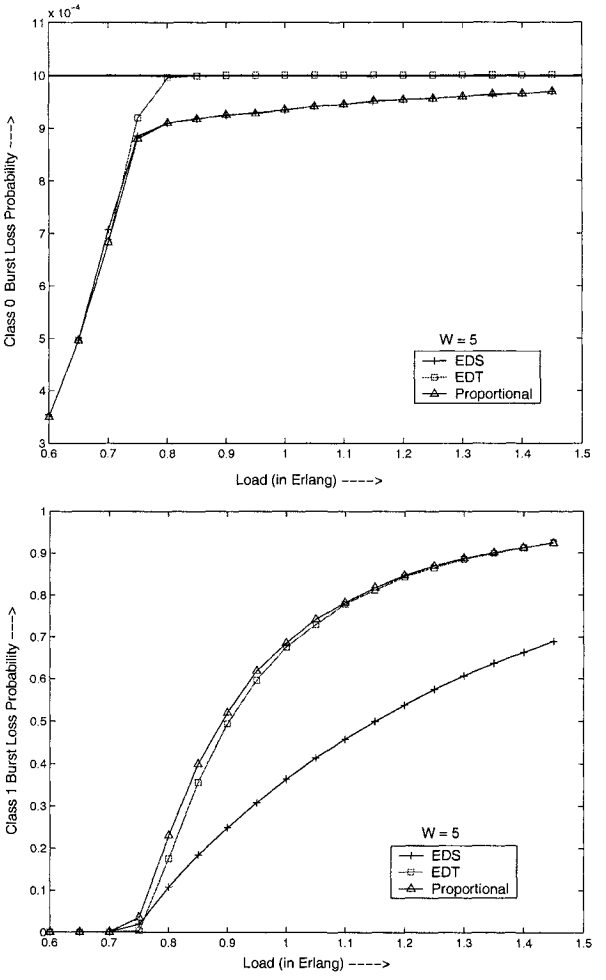


Figure 7.10. (a) Class 0 and (b) Class 1 loss probability versus load for EDS, EDT and Proportional schemes.

We now describe an approach to assign labels and provision the necessary wavelengths for the integrated schemes, using a two-class example. Fig. 7.11 presents the burst scheduling process in the integrated schemes. EDS is implemented by an *EDS Labeler*, and wavelength grouping is implemented by a *WG Scheduler*. Initially, the EDS labeler labels each burst according to the class of the burst and the value of the corresponding early dropping vector, $ED_1 = \{e_1\}$. As shown in Table I, a burst is assigned a Label $L0$, either if the burst is of Class 0, or if the burst is of Class 1 and e_1 is 0. A burst is assigned a Label $L1$ if the

burst is of Class 1 and e_1 is 1. The labeled burst is then sent to the WG scheduler, which schedules the burst solely based on its label. Table II gives the number of wavelengths provisioned for each group of bursts with a given label. A burst labeled $L0$ can be scheduled on any of the W wavelengths. This is because, when the early dropping scheme is not triggered, all the arriving bursts are labeled $L0$, resulting in all the wavelengths being utilized. A burst labeled $L1$ can only be scheduled on W_{C1} wavelengths, where $W_{C_1} = W - W_{C_0}$. This restriction ensures that there are a required number of W_{C0} wavelengths on which the bursts labeled $L0$ can be scheduled.

Integrated EDS and SWG

Using SWG, a burst with Label $L0$ can be scheduled on any available wavelength. while a burst with Label $L1$ can only be scheduled on the statically pre-assigned W_{C_1} wavelengths. Figure 7.11 illustrates three possible burst arrival scenarios. The current wavelength allocation is shown on the right hand side of the figure. In Case 1, when a Class 0 burst with Label $L0$ arrives, the burst is scheduled on Wavelength 2. In Case 2, when a Class 1 burst with Label $L0$ arrives, the burst is also scheduled on Wavelength 2. While in Case 3, when a Class 1 burst with Label $L1$ arrives, the burst cannot be scheduled on Wavelength 2, since a burst labeled $L1$ can be scheduled only on the statically provisioned Wavelength 3.

Integrated EDS and DWG

Following LAUC, the DWG scheduler records the label of the latest-scheduled burst on every wavelength. When a burst labeled $L0$ arrives, DWG can schedule the burst on any available wavelength. On the other hand, when a burst with Label $L1$ arrives, the burst is scheduled on any of the available wavelengths, as long as the number of bursts labeled $L1$ already scheduled at the arrival time of the arriving burst is less than W_{C_1}. In Fig. 7.11, suppose the label of the latest scheduled burst recorded on each of Wavelengths 0, 1, and 3 is $L0$. With the DWG scheduler, for all three burst arrival scenarios, the arriving burst can be scheduled on Wavelength 2.

The integrated schemes provide better resource allocation compared to each of the stand-alone schemes for the following reasons. First, in the wavelength grouping schemes, the Class 1 bursts can be scheduled only on W_{C_1} wavelengths, while, in the integrated schemes, the Class 1 bursts with Label $L0$ can be scheduled on any wavelength. Second, compared to early dropping schemes, the integrated schemes reduce the

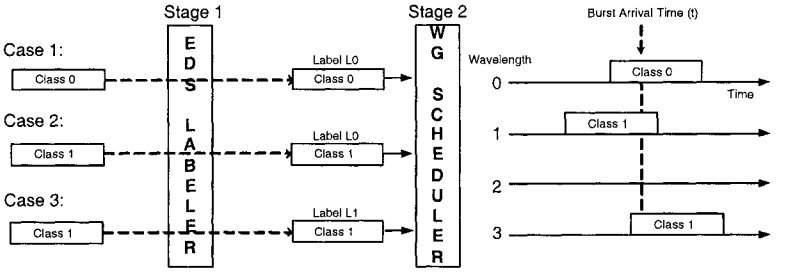


Figure 7.11. Illustration of the integrated schemes.

unnecessary intentional dropping of Class 1 bursts, since the Class 1 bursts with Label $L1$ can use a maximum of W_{C_1} wavelengths.

References

[1] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. *ACM Computer Communication Review*, pages 3–12, 1989.

[2] C. Dovrolis and P.Ramanathan. A case for relative differentiated services and the proportional differentiation model. *IEEE Network*, October 1999.

[3] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. *IEEE/ACM Transactions on Networking*, 10(1):12–26, February 2002.

[4] C. Dovrolis and P. Ramanathan. Dynamic class selection: From relative differentiation to absolute QoS. In *Proceeding, IEEE ICNP*, pages 120–128, November 2001.

[5] Y. Chen, M. Hamdi, D.H.K. Tsang, and C. Qiao. Proportional differentiation - a scalable QoS approach. In *IEEE Communications Magazine*, June 2003.

[6] M. Yoo and C. Qiao. Supporting multiple classes of service in IP over WDM networks. In *Proceedings, IEEE Globecom*, pages 1023–1027, December 1999.

[7] M. Yoo, C. Qiao, and S. Dixit. QoS performance of optical burst switching in IP-over-WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2062–2071, October 2000.

[8] Y. Chen, M. Hamdi, and D.H.K. Tsang. Proportional QoS over OBS network. In *Proceedings, IEEE Globecom*, volume 3, pages 1510–1514, November 2001.

[9] F. Poppe, K. Laevens, H. Michiel, and S. Molenaar. Quality-of-service differentiation and fairness in optical burst-switched networks. In *Proceedings, SPIE OptiComm*, volume 4874, pages 118–124, July 2002.

[10] V. M. Vokkarane and J. P. Jue. Prioritized routing and burst segmentation for QoS in optical burst-switched networks. In *Proceedings, Optical Fiber Communication Conference (OFC)*, volume WG6, pages 221–222, March 2002.

[11] M. Yang, S.Q. Zheng, and D. Verchere. A QoS supporting scheduling algorithm for optical burst switching DWDM networks. In *Proceedings, IEEE Globecom*, volume 4, November 2001.

[12] C-H Loi, W. Liao, and D-N Yang. Service differentiation in optical burst switched networks. In *Proceedings, IEEE Globecom*, volume 3, pages 2313–2317, November 2002.

[13] Y. Xiong, M. Vanderhoute, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1838–1854, October 2000.

[14] V.M. Vokkarane, Q. Zhang, J.P. Jue, and B. Chen. Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks. In *Proceedings, IEEE Globecom*, volume 3, pages 2747–2751, November 2002.

[15] V. M. Vokkarane and J. P. Jue. Prioritized burst segmentation and composite burst assembly techniques for QoS support in optical burst switched networks. *IEEE Journal on Selected Areas in Communications*, 21(7):1198–1209, September 2003.

[16] F. Farahmand and J. P. Jue. A preemptive scheduling technique for OBS networks with service differentiation. In *Proceedings, IEEE Globecom*, December 2003.

[17] J. Liu and N. Ansari. Forward resource reservation for QoS provisioning in OBS systems. In *Proceedings, IEEE Globecom*, volume 3, pages 2777–2781, December 2003.

[18] E. Kozlovski, M. Dueser, A. Zapata, and P. Bayvel. Service differentiation in wavelength-routed optical burst-switched networks. In *Proceedings, Optical Fiber Communication Conference (OFC)*, pages 774–775, March 2002.

[19] E. Kozlovski and P. Bayvel. QoS performance of WR-OBS network architecture with request scheduling. In *Proceedings, IFIP Conference on Optical Network Design and Modeling (ONDM)*, February 2002.

[20] I. de Miguel, E. Kozlovski, and P. Bayvel. Provision of end-to-end delay guarantees in wavelength-routed optical burst-switched networks. In *Proceedings, IFIP Conference on Optical Network Design and Modeling (ONDM)*, February 2002.

[21] E. Kozlovski, M. Dueser, I. de Miguel, and P. Bayvel. Analysis of burst scheduling for dynamic wavelength assignment in optical burst-switched networks. In *IEEE Lasers & Electro-Optics Society (LEOS)*, page TuD2, November 2001.

[22] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson. Signaling support for multicast and QoS within the JumpStart WDM burst switching architecture. *SPIE Optical Networks Magazine*, 4(6):68–80, November/December 2003.

[23] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANs. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.

[24] L. Yang, Y. Jiang, and S. Jiang. A probabilistic preemptive scheme for providing service differentiation in OBS networks. In *Proceedings, IEEE Globecom*, pages 2689–2673, December 2003.

[25] H. C. Cankaya, S. Charcranoon, and T. S. El-Bawab. A preemptive scheduling technique for OBS networks with service differentiation. In *Proceedings, IEEE Globecom*, pages 2704–2708, December 2003.

[26] Q. Zhang, V.M. Vokkarane, B. Chen, and J.P. Jue. Absolute QoS differentiation in optical burst-switched networks. In *Proceedings, IEEE Globecom*, volume 5, pages 2628–2632, December 2003.

Chapter 8

OTHER TOPICS

As optical burst switching moves closer to reality, research focus will begin to shift from optical burst switching layer protocols and architectures to the interactions between optical burst switching and higher-layer protocols and applications. Further effort will also be directed towards the development of practical implementation of optical burst switched networks in testbed deployments.

8.1 Labeled OBS

When optical burst switching is eventually deployed, it is likely to provide transport services for higher-layer protocols such as IP. Thus, it is important to determine how an optical burst-switched network will interact with the IP layer. If IP is deployed over an optical burst-switched network, the two layers can either be implemented independently of one another, such that each layer with its own control and management mechanisms, or the two layers can be implemented in an integrated manner in which a common control plane is shared by the two layers.

In order to reduce management costs, it is possible to implement optical burst switching within the framework of generalized multiprotocol label switching (GMPLS). In GMPLS, virtual-circuit paths are established in the network through the use of labels. These paths are referred to as label-switched paths (LSPs). Each node in the network, referred to as a label-switched router (LSR), maintains a forwarding table which specifies, for each label on each incoming port, the appropriate outgoing label and outgoing port.

The establishment of an LSP requires the maintenance and distribution of topology and state information, a method for determining the route of LSPs, and a signaling protocol for establishing and maintaining

LSPs. Typically, in IP networks, state information is maintained by the open shortest path first (OSPF) protocol. The OSPF protocol maintains topology and link-state information by periodically sending hello messages to neighbors and by periodically flooding link-state advertisements (LSAs) throughout the network. Extensions to the OSPF protocol for supporting GMPLS in WDM optical networks have been proposed [1].

Routing in GMPLS can be either *hop-by-hop routing* or *explicit routing*. In hop-by-hop routing, the route for an LSP is determined on a hop-by-hop basis, with each node only knowing the next hop node in the path. Once an LSP is established in this manner, all data with the corresponding label will follow the same path. In explicit routing, routes for LSPs are determined by a centralized entity, such as the source node (source routing), according to certain policies or traffic engineering objectives. Policies may be based on metrics such as path length or link congestion. Explicit routing approaches typically require that the routing entity has some knowledge of the network topology and link-state information.

The signaling for establishing LSPs in GMPLS can be done through protocols such as the constraint-based label distribution protocol (CR-LDP), or the resource reservation protocol with traffic engineering extensions (RSVP-TE). Both protocols send control messages along the selected route in an attempt to reserve resources and to configure the label forwarding tables at each label-switched router.

Once a LSP is established between a source node and a destination node, the source node will apply the appropriate label to the incoming data, and the data will be forwarded along the LSP. In packet-switched networks, each packet is assigned a label at the ingress node, and is routed through the network along a pre-determined label-switched path. In circuit-switched WDM optical networks, labels correspond to wavelengths, and LSPs correspond to lightpaths. In this case, incoming packets are assigned a given wavelength and sent on the appropriate output port. The packet will traverse the lightpath end-to-end entirely in the optical domain.

The concept of GMPLS and label switching can also be extended to optical burst-switched networks [2]. In this approach, referred to as labeled optical burst switching (LOBS), a labels are applied to the burst header packets. Each optical burst switching node is considered as a label-switched router and will appropriately route incoming bursts and swap labels in the burst header packets. The use of label switching in an OBS network enables traffic engineering by allowing the establishment of explicit and constraint-based routes for bursts.

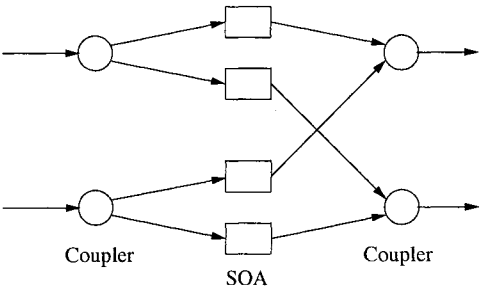


Figure 8.1. Semiconductor optical amplifier (SOA) switch.

The use of explicit routes in LOBS networks to provide load-balanced routing is investigated in [3]. The authors formulate an integer linear program with the objective of either minimizing the number of links whose utilization is above a given threshold or minimizing the total bandwidth consumed on links whose utilization is above a given threshold.

8.2 Multicasting in OBS

One area of practical interest in networks is multicasting, which is the transmission of data from one or more sources to many destinations. In optical networks, multicasting can either be supported through the optical splitting of a signal or through electronic duplication of data. All-optical multicasting in optical burst-switched networks requires the use of optical splitters at nodes. An example of a node capable of multicasting capabilities is the SOA-based switch shown in Fig. 8.1.

In an optical burst-switched network, multicasting can be implemented by sending multiple unicast bursts or by sending a burst along a multicast tree [4]. In the multiple unicast approach, a separate copy of a given burst is sent to each of the multicast destinations. The multiple unicast approach is simple and does not require optical splitters at each node. Instead, electronic duplication is required at the source node. The disadvantage of the multiple unicast approach is that it is not efficient in terms of bandwidth utilization. In the multicast tree approach, each multicast session can either have its own specific multicast tree, or multicast sessions may share a set of multicast trees. In the case of individual multicast trees, a minimum-cost tree that connects the source to the destinations should be found in order to minimize the resources consumed by the multicast transmission. The limitation of individual multicast trees is that, if a multicast session does not consist of much traffic, then the bursts that are transmitted over the multicast tree will

be small, resulting in high overhead. In order to reduce overhead, the multicast traffic must be combined and assembled into the same bursts as other traffic.

In the tree-sharing multicast approach presented in [4], the set of multicast sessions originating from a given source are partitioned into subsets, called multicast sharing classes (MSCs). Each MSC shares a single multicast tree. A simple strategy for grouping multicast sessions into MSCs is to group all sessions that have the same set of destination nodes. In this case, a single tree that spans all destinations in the multicast set is sufficient. Another strategy is to include all multicast sessions for which the multicast set is a subset of the destination nodes in a given multicast session. In this case, a tree that spans all destinations in the given multicast session will also span all nodes in multicast sessions whose multicast set consists of a subset of those destinations. A more general approach is to group multicast sessions whose destination sets have sufficient overlap. In the latter two cases, there is some degree of bandwidth inefficiency, since bursts or packets within a burst may end up going to nodes that are not a part of a given multicast set. However, by sharing the tree, bursts will be longer, leading to less overhead. The work in [5] extends the concept of shared multicast trees to the case in which nodes may dynamically join and leave multicast sets. In this case, the MSCs must be updated over time.

In [6], the problem of supporting reliable multicasting in OBS networks is considered. Typically, if a burst belonging to a multicast transmission is lost, then the lost data would need to be retransmitted by the higher-layer reliable IP multicasting protocol or by the TCP layer. Handling losses at higher layers may result in a larger number of duplicated transmissions and higher end-to-end delay. A more efficient approach is to support burst recovery in the OBS network itself. In this approach, if a multicast burst is dropped by a node, that node will send a negative acknowledgement (NAK) towards the source node along the multicast tree. When the NAK reaches the first upstream branching node (including nodes at which the burst is split optically as well as nodes at which the burst is received electronically), the branching node sends a retransmission request to the closest multicast member node which has successfully received the burst. All branching nodes along the tree are required to maintain state information for each multicast burst.

8.3 Protection for Optical Burst-Switched Networks

An important issue in optical networks is survivability. When a link or node fails, the network should have the capability to continue carrying

critical traffic. In traditional circuit-switched optical networks, survivability is provided through protection and restoration mechanisms. Protection mechanisms allocate and reserve spare backup resources prior to failure. When a failure occurs, traffic is switched to the backup resources. In restoration, no resources are reserved in advance. Instead, resources are discovered and reserved after the failure occurs. Protection schemes provide a higher degree of survivability, but consume a greater amount of resources. Restoration schemes, on the other hand, do not consume much resources, but do not guarantee that there will be enough capacity to handle traffic in the case of a failure.

In a labeled optical burst-switched network, survivability schemes may be required to prevent burst losses in the event of link failures. If there are connections in the form of LSPs over the failed link, then any bursts associated with those LSPs will be lost if no action is taken. A fairly straightforward approach to handling link failures is to simply use deflection. Once a node determines that a link has failed, it deflects any bursts headed for that link to a different link. This approach is simple and requires only a local decision. However, in order to implement deflection within a labeled optical burst switching environment, labels would need to be distributed to nodes other than those on the route of the LSP.

Protection schemes in optical burst-switched networks can also be provided through the establishment of redundant LSPs [2]. With labeled optical burst switching, a backup LSP can be established for each working LSP, thereby providing dedicated path protection. A source node initially sends bursts over only one of the LSPs. When a failure occurs, the source node is notified of the failure and begins sending the bursts over the backup LSP. In this scheme, no additional resources are required in the network, other than the LSP entries at label-switched routers along the backup path. The technique can also be extended such that a primary LSP is protected by multiple backup LSPs, with each backup LSP capable of carrying a fraction of the traffic from the original LSP.

In [7], the authors propose a 1+1 protection architecture for optical burst-switched networks. This approach is based on MPLS 1+1 protection in which two disjoint label-switched paths are established between the ingress and egress label-switching router. The ingress node duplicates incoming packets and sends one copy on each of the label-switched paths. In 1+1 optical burst switching protection, two disjoint LSP routes are determined for each burst session. All bursts belonging to a given session will be copied and sent out on both LSPs. Thus, if a link fails on one LSP, then bursts will continue to be received on the other LSP.

The advantage of $1 + 1$ protection is that no additional actions are required in the optical burst-switched network in order to recover from a failure. The disadvantage is that the scheme uses at least twice as many resources as the unprotected case. Also, the destination node must be able to eliminate redundant bursts.

8.4 TCP over OBS

Increasing attention is being given to the interaction of higher-layer protocols with OBS. In particular, the effects of burst assembly and losses in the OBS layer can have a significant impact on TCP performance.

TCP is intended to provide a reliable transport layer over an unreliable network layer. TCP includes mechanisms for acknowledging received data and resending data that is lost. It also provides a flow/congestion control mechanism that reduces the sending rate if congestion is detected in the network. Several versions of TCP have been proposed and implemented. The more popular versions include TCP Reno, TCP New Reno, and TCP SACK.

8.4.1 TCP Reno, New Reno, and SACK

In TCP Reno, the TCP source maintains a variable CW, which indicates the size of the congestion window. The congestion window is used to determine the maximum number of unacknowledged segments the TCP sender can have.

TCP Reno has two mechanisms for detecting the loss of data. These mechanisms are triple-duplicate ACKs and timeouts. A triple-duplicate ACK is triggered when the TCP source receives three duplicate ACKs for the same segment. The TCP sender interprets a triple duplicate ACK event as an indication that one or more segments have been lost to light congestion. The TCP sender will halve its congestion window size and immediately retransmit one lost segment, a procedure known as *fast retransmission*. After resending the segment, the TCP source enters a *fast recovery* phase. In this phase, the TCP source will increase its congestion window size by one for each duplicate ACK that it receives. After receiving half a window of duplicate ACKs, the congestion window size will be the same as the window size prior to the TD detection. Thus, the source can send a new packet for each additional duplicate ACK that it receives. The source exits fast recovery upon the receipt of the ACK that acknowledges the retransmitted lost segment, and enters into a *congestion avoidance* phase.

A timeout event occurs when a TCP source does not receive an acknowledgement for a segment within a certain timeout duration. Typically this timeout duration is on the order of some multiple of the round-trip propagation delay. Loss due to a timeout event indicates that there is heavy congestion in the network. The TCP source will respond by retransmitting the lost segment and entering a *slow start* phase. In the slow start phase, the TCP source sets its congestion window size to one, and increases the congestion window by one for each acknowledgement that it receives. Once the congestion window size reaches a certain threshold, the TCP source enters the congestion avoidance phase.

A limitation of TCP Reno is that, if multiple segments are lost, a triple duplicate ACK event will be triggered for each lost segment, resulting in the window size being halved for each of these events. If the window size becomes less than three, it will not be possible to receive a triple duplicate ACK, and any further loss will result in a timeout event. This timeout event will cause the TCP source to enter the slow start phase.

TCP New Reno attempts to overcome some of the limitations of TCP Reno by using *partial ACKs*. A partial ACK is an ACK that acknowledges a new segment, but not the segment with the highest sequence number when fast recovery was triggered. When a triple duplicate ACK is received, the TCP source retransmits one lost segment and enters the fast recovery phase. When a new ACK is received, if the ACK is not for the segment that was already retransmitted and is not for the segment with the highest outstanding sequence number, then the ACK is considered to be a partial ACK. In this case, the TCP source immediately retransmits the lost segment indicated by the partial ACK without waiting for the arrival of three duplicate ACKs. If the ACK acknowledges the segment with the highest sequence number, then the TCP source will exit the fast recovery phase and will enter into the congestion avoidance phase. While TCP New Reno can prevent the source from entering the slow start phase when multiple segments are lost, it can still result in a lengthy retransmission period during which no new segments can be sent.

TCP SACK extends TPC Reno by including more information in the ACK. The ACK contains a number of SACK blocks, where each SACK block specifies a non-continuous set of packets that has been received and queued at the receiver side. When a triple duplicate ACK loss is detected, the TCP source retransmits one lost segment and enters the fast recovery phase. The TCP source selectively retransmits one lost segment that is reported by a SACK block for each partial ACK it receives. When an ACK acknowledges the highest sequence number sent when fast retransmission was triggered, TCP SACK exits the fast recovery

phase and enters congestion avoidance. By giving the SACK information, the sender can avoid unnecessary delays and retransmissions as in Reno and New-Reno, resulting in improved throughput.

8.4.2 TCP over OBS

When TCP is implemented over an optical burst-switched network, a burst loss may result in the loss of several TCP segments, which may be interpreted as heavy congestion by the TCP source. However, the loss of a single burst does not necessarily indicate congestion in the optical burst-switched network. If the loss of a single burst in the optical burst-switched network leads to a timeout event at the TCP source, and if the optical burst-switched network is not congested, then this timeout event is referred to as a false timeout (FTO) [8]. In such a situation, entering slow start is not desirable, since doing so would unnecessarily reduce the TCP throughput. Several mechanisms for detecting FTOs and avoiding slow start are presented in [8].

In the first method, the TCP source must estimate how many of its segments will be included in the same burst. If the congestion window size is less than the estimated burst size, then a timeout event is treated as a false timeout. In this case, all of the segments within a window are likely to be contained in a single burst. Thus, a burst loss would always result in a timeout event regardless of whether or not there is congestion in the optical burst-switched network. If the congestion window size is greater than the estimated burst size, then a timeout event is treated as a true timeout. In this case, the segments in a given window are likely to be spread over more than one burst, and a timeout event will occur only if all of these bursts are lost. The loss of multiple bursts is likely to be a sign of congestion in the optical burst-switched network.

A second approach proposed in [8] is for the OBS ingress node to inform the TCP source of which TCP segments are included in each burst. When a timeout event occurs, the TCP source can immediately determine whether all segments were in the same burst or not. If all segments were in the same burst, then the timeout is treated as a false timeout event. This approach requires the OBS layer to be aware of TCP segments.

In a third approach, each burst header packet contains information on the TCP segments contained within the burst. When the burst is dropped, the dropping node will examine the header and send a negative acknowledgement (NAK) to the TCP source, indicating which TCP segments were lost. If the TCP source determines that all segments in a congestion window were contained within the same lost burst, then it will interpret a timeout event as a false timeout. If all segments in the

congestion window were not contained in the lost burst, then a timeout event will be interpreted as a true timeout.

The advantage of detecting a false timeout is that the TCP source can avoid entering the slow start phase if a timeout event is caused by a single burst loss rather than by network congestion. A disadvantage of the second and third approaches is that the OBS layer needs to know about TCP segments, and the TCP layer needs to be aware of bursts.

8.5 OBS Testbeds

A number of OBS testbeds have been developed and deployed in laboratory settings. These testbeds are intended to demonstrate the feasibility of the concept of OBS and to test various OBS protocols.

8.5.1 TIPOR

Although optical burst switching can be implemented over any all-optical switching technology, a number of switch and router testbeds have been developed specifically with optical burst switching in mind. One such project is the TIPOR (Terabit IP optical router) project developed at Alcatel [9]. In this testbed, IP packets or ATM cells are assembled into bursts at the router inputs, switched as bursts through an optical fabric, and disassembled into individual packets or cells at the router outputs. Thus, burst switching is just carried out across the router rather than across a network.

The optical switching fabric is based on a broadcast and select architecture in which semiconductor optical amplifiers are used to select signals for a given output. The architecture also makes use of packet mode receivers that are capable of receiving bursts at data rates of up to 10 Gb/s and recovering the clock within 12 ns.

8.5.2 JumpStart

JumpStart is an OBS project [10] that is being developed by North Carolina State University and MCNC. JumpStart specifies a JIT-based architecture for OBS networks.

The JIT-based architecture defined in the JumpStart project was implemented over the ATDnet all-optical networking testbed in the Washington DC area [11]. The ATDnet testbed consists of several sites interconnected by optical WDM fiber links, with each wavelength operating at OC-48 data rates. Each site maintains a Firstwave SIOS 1000 MEMs-based optical crossconnect for all-optical switching.

The JIT implementation over the ATDnet testbed involved the installation of JIT OBS network controllers, referred to as JITPACs (Just-in-

Time Protocol Acceleration Circuit), at three of the ATDnet sites. Each JITPAC consisted of a Motorola MPC8260 PowerPC processor, an Altera EP20K400C FPGA, 4 MB of SDRAM, a 64 MB SDRAM DIMM module, 16 MB of flash ROM, two serial ports, an ATM interface for the signaling channel, and an Ethernet interface for controlling the OXC. The cost of each JITPAC was approximately $4,000.

The JIT protocol defined in the JumpStart project utilizes immediate reservation and either implicit or explicit release. The protocol supports both analog and digital formats for data bursts and also supports multicasting.

References

[1] K. Kompella and Y. Rekhter. Osfp extensions in support of generalized MPLS. *draft-ietf-ccamp-ospf-gmpls-extensions-12.txt*, October 2003.

[2] C. Qiao. Labeled optical burst switching for IP-over-WDM integration. *IEEE Communications Magazine*, 38(9):104–114, September 2000.

[3] J. Zhang, H.-J. Lee, S. Wang, X. Qiu, K. Zhu, Y. Huang, D. Datta, Y.-C. Kim, and B. Mukherjee. Explicit routing for traffic engineering in labeled optical burst-switched WDM networks. In *To appear, Proceedings, ICCS*, 2004.

[4] M. Jeong, Y. Xiong, H. C. Cankaya, M. Vandenhoute, and C. Qiao. Efficient multicast schemes for optical burst-switched WDM networks. In *icc*, pages 1289–1294, June 2000.

[5] M. Jeong, C. Qiao, Y. Xiong, and M. Vandenhoute. Bandwidth-efficient dynamic tree-shared multicast in optical burst-switched networks. In *icc*, pages 630–636, June 2001.

[6] M. Jeong, C. Qiao, and Y. Xiong. Reliable WDM multicast in optical burst-switched networks. *onm*, 2(2):29–40, March/April 2000.

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

[8] X. Yu, C. Qiao, and Y. Liu. TCP implementations and false time out detection in OBS networks. In *Proceedings, IEEE Infocom*, March 2004.

[9] F. Masetti, D. Zriny, D. Verchere, and et al. Design and implementation of a multi-terabit optical burst/packet router prototype. In *ofc*, March 2002.

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

[11] 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 *globecom*, December 2003.

Index