DESIGN AND ANALYSIS OF ARCHITECTURES AND PROTOCOLS FOR OPTICAL BURST-SWITCHED NETWORKS

by

VINOD MANDAYAM VOKKARANE, B.E., M.S.

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

August 2004

DESIGN AND ANALYSIS OF ARCHITECTURES AND PROTOCOLS FOR OPTICAL BURST-SWITCHED NETWORKS

Publication No. _____

Vinod Mandayam Vokkarane, Ph.D.
The University of Texas at Dallas, 2004

Supervising Professor: Dr. Jason Jue

Current fast-growing Internet traffic is demanding increased network capacity each day, as well as support for differentiated services. Wavelength-division multiplexing (WDM) technology has provided an opportunity to drastically increase network capacity, while optical burst switching offers all-optical, high-speed, format-transparent switching, which is an essential characteristic for future networks that need to support different classes of data.

In this report, we analyze several critical issues affecting optical burst-switched networks, such as contention resolution, channel scheduling, burst assembly, signaling, and quality of service (QoS).

We introduce a new approach called *Burst Segmentation*, to reduce packet loss during contention resolution. We propose non-preemptive and preemptive scheduling algorithms that use burst segmentation to resolve contention, so as to achieve lower packet loss.

We also investigate the handling of prioritized data traffic. Our first approach for providing QoS support is by introducing *prioritized burst segmentation* in the network core network. The prioritized burst segmentation scheme allows high-priority bursts to preempt low-priority bursts and enables full class isolation between bursts of different priorities. In the second

approach for providing QoS, we introduce a new technique for assembling packets into a burst referred to as *composite burst assembly*. In this technique, a composite burst is created by combining packets of different classes into the same burst. We describe a generalized burst assembly framework, and we propose several composite burst assembly methods. In the third approach for supporting QoS, We propose a *differentiated threshold-based burst assembly* scheme to provide QoS in optical burst-switched networks. Through simulations, we also show the presence of an optimal threshold value of minimizes packet loss for given network parameters.

We also develop a generalized signaling framework for optical burst-switched networks, and we propose technique called *intermediate node initiated* (INI) signaling, for optical burst-switched networks. INI can provide different levels of loss and delay characteristics based on the client applications requirements.

In general, all the proposed solutions solve many of the fundamental issues faced by optical burst-switched networks, thereby making OBS more practical and deployable in the near future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Evolution of Optical Networks

Over the last decade, the field of networking has experienced growth at a tremendous rate. The popularity of the Internet is soaring as more people gain an increased awareness of the vast amounts of information available at the click of a button. This explosion is leading to many new opportunities in networking, as people demand faster and better applications and services, such as World Wide Web browsing, video-on-demand, and interactive television. 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. We need to be able to scale current networks to support the increasing volumes of information.

Optical networks are a logical choice to meet future communication demands, with optical fiber links offering huge bandwidths on the order of 25 THz. In order to meet these growing needs, optical wavelength-division multiplexing (WDM) communication systems have been deployed in many telecommunications backbone networks. In WDM networks, channels are created by dividing the bandwidth into a number of wavelength or frequency bands, each of which can be accessed by the end-user at peak electronic rates [1]. In order to efficiently utilize this bandwidth, we need to design efficient transport architectures and protocols based on state-of-the-art optical device technology.

Figure 1.1 shows the evolution of the different optical transport methodologies [2]. The first generation optical network architectures consist of *point-to-point WDM links*. Such

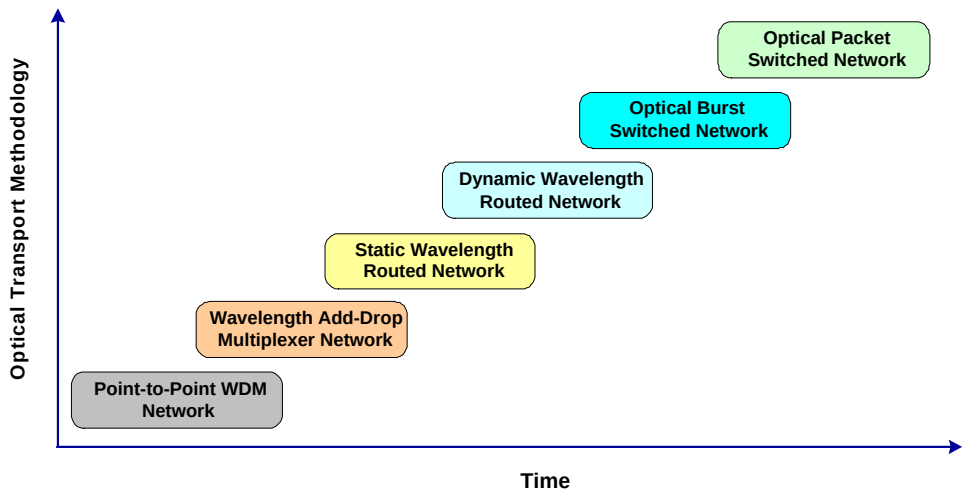


Figure 1.1. Evolution of optical transport methodologies.

networks are comprised of several point-to-point links, at which all traffic coming into each node from an input fiber is dropped and converted from optics to electronics, and all outgoing traffic has to be converted back from electronics to optics before being sent on the outgoing fiber. This dropping and adding i of the entire traffic at every node in the network incurs significant overhead in terms of switch complexity and data transmission 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 add-drop devices can be used.

Second-generation optical network architectures are based on wavelength add-drop multiplexers (WADM) [3], where traffic can be added and dropped at the WADMs location. WADMs can terminate only selected channels from the fiber and let 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 WADM, we can reduce overall cost by dropping only the wavelengths whose final destination is same as the current node, and allowing all other wavelengths to bypass the node. WADMs can serve as a basis for switching, wherein the WADMs is remotely configured to drop any wavelength to any port without manual intervention. We can perform circuit, or point-to-

point, switching in the optical domain with a WADM. The WADMs are mainly used to build optical WDM ring networks which are expected to deployed mainly in the metropolitan area market.

In order to build a mesh network consisting of multi-wavelength fiber links, we need appropriate fiber interconnection devices. Third-generation optical network architectures are based on all-optical interconnection devices. These devices fall under the following three broad categories, namely *passive star*, *passive router*, and *active switch* [4]. The *passive star* is a broadcast device. A signal that is inserted on a given wavelength from an input fiber port will have its power equally divided among (and appear on the same wavelength on) all output ports. A *passive router* can separately route each of several wavelengths incident on an input fiber to the same wavelength on separate output fibers. The *active switch* also allows wavelength reuse, and it can support simultaneous connections through itself. The passive star is used to build local WDM networks, while the active switch is used for constructing wide-area wavelength-routed networks. The passive router has mainly found application as a mux/demux device.

In this dissertation, we focus on optical wide-area (long-haul) mesh network architectures and we are primarily concerned with transport methodologies based on *optical cross connects (OXC)* (or active switches). There are primarily three all-optical transport methodologies, namely, wavelength routing (circuit-switched), optical burst switching, and optical packet switching. We describe each of the transport methodologies below. Note that all-optical transport methodologies are characterized by a bufferless core network, so as to benefit from the high optical data transmission rates.

In wavelength routed WDM networks, end users communicate with one another via all optical WDM channels, which are referred to as lightpaths [5]. A lightpath is used to support a connection in a wavelength routed WDM network and may span multiple fiber links. In the absence of wavelength converters, a lightpath must occupy the same wavelength

on all the fiber links through which it traverses. This property is known as the *wavelength continuity constraint*.

Given a set of connections, the problem of setting up lightpaths by routing and assigning a wavelength to each connection is called 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 then 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 have been extensive study to solve both the static and the dynamic RWA problems [6].

Wavelength-routed connections are fairly static and they may not be able to accommodate the highly variable and bursty nature of Internet traffic in an efficient manner. In order to meet the growing bandwidth demands in a metropolitan or a long-haul environment, transport methodologies that support fast resource provisioning and that handle bursty traffic must be developed. Also, the rapid increases in data traffic suggest that all-optical WDM networking technologies, capable of switching at sub-wavelength granularity, are attractive for meeting diverse traffic demands of the next-generation networks. Optical Burst Switching (OBS) and Optical Packet Switching (OPS) are two such promising methods for transporting traffic directly over a bufferless optical WDM network [7, 8, 9, 10].

Optical packet switching is capable of dynamically allocating network resources with fine packet-level granularity while offering excellent scalability [8, 11, 12, 13, 14]. In an optical packet-switched network, individual photonic switches are combined to form a network. Packets can arrive at the input ports of each node at different times. In packet-switched

networks, bit-level synchronization and fast clock recovery are required for packet header recognition and packet delineation.

Optical packet-switched networks can be classified into two categories: slotted (synchronous) and unslotted (asynchronous) networks. In a slotted network, all the packets have the same size. Packets are placed together with the header inside a fixed time slot, which has a longer duration than the packet and header to provide guard time. In an unslotted network, the packets may or may not have the same size, and the packets arrive and enter the switch without being aligned. Therefore, the packet-by-packet switch action could take place at any point in time. This can leads to contention of different incoming packets for the same outgoing resource. Obviously, in unslotted networks, the chance for contention is larger because the behavior of the packets is more unpredictable and less regulated. On the other hand, unslotted networks are easier and cheaper to build, more robust, and more flexible compared to slotted networks.

A possible near-term alternative to all-optical circuit switching and all-optical packet switching is *optical burst switching* [10]. 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.

Circuit and packet switching have been used for many years for voice and data communications respectively. Burst switching [15, 16, 17], on the other hand, is less common. Switching techniques primarily differ based on whether data will use *switch cut-through* or *store and forward*. In circuit switching, a dedicated path between two stations is necessary. A call is established, the data is transferred, and the call is disconnected. Resource reservation is done for the duration of the call. In packet switching, the data is broken into small packets and transmitted. The resources can be shared by different sources. End stations can send

and receive data at their own speed. The individual packets can be individually switched or a virtual circuit can be set up. In the first case, the routing decision is done at a packet level while in the later, it is on a virtual channel level. Individual routing may lead to out-of-order message delivery.

Circuit switching is advantageous when we have constant data rate traffic (fixed delays) in the network, like voice traffic; however, it is not suitable under bursty traffic conditions, or when circuits are idle [18]. Packet switching works well with variable-rate traffic, like data traffic, and can achieve higher utilization. Prioritization of data can also be incorporated in packet switching; however, it is difficult to give QoS assurances (best effort service), and packets can have variable delays [7].

Optical burst switching was introduced only recently for optical (WDM) networks, and is thus not as well understood as optical circuit and packet switching. Circuit switching uses two-way reservation schemes that have a large round trip. Packet switching has a large buffer requirement, complicated control, and strict synchronization issues. OBS is designed to achieve a balance between the coarse-grained circuit switching and the fine-grained packet switching. As such, a burst may be considered as having an intermediate "granularity" as compared to circuit and packet switching. OBS uses one-way reservation schemes with immediate transmission, in which the data burst follows a corresponding packet without waiting for an acknowledgment [19, 20, 21, 22, 23, 24]. Optical burst switching techniques differ based on how and when the network resources, like bandwidth, are reserved and released.

Optical burst switching is an adaptation of an International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) standard for burst switching in asynchronous transfer mode (ATM) networks, known as ATM block transfer (ABT) [25]. There are two versions of ABT: ABT with delayed transmission and ABT with immediate transmission. In the first case, when a source wants to transmit a burst, it sends a packet to the ATM switches on the path of the connection to inform them that it wants to transmit a burst. If all the switches on the path can accommodate the burst, the request is accepted and the

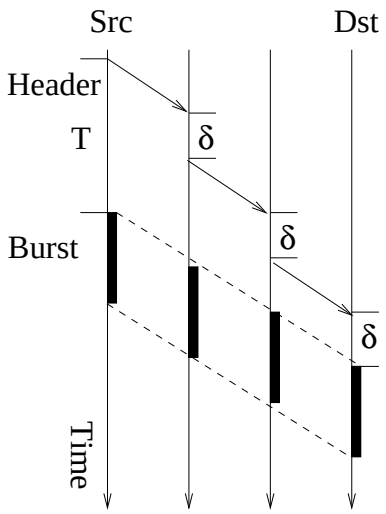


Figure 1.2. The use of offset time in OBS.

source is allowed to go ahead with its transmission. Otherwise, the request is refused, and the source has to send another request later. In ABT with immediate transfer, the source sends the request packet, and then immediately following the request, without receiving a confirmation, the source transmits its burst. If a switch along the path cannot carry the burst due to congestion, the burst is dropped. These two techniques have been adopted to optical networks.

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 1.2) 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. Hence, the OBS paradigm supports dynamic bandwidth allocation and statistical multiplexing of data, efficiently utilizing the WDM links.

The common signaling schemes for reserving resources in OBS networks are tell-and-go (TAG), tell-and-wait (TAW), and just-enough-time (JET). The TAG scheme [26, 27] is similar to the ABT with immediate transmission, and the TAW scheme [27] is similar to ABT with delayed transmission. An intermediate scheme known as JET was proposed in [10].

In the TAG scheme, the source transmits the control packet and then immediately transmits the optical burst. In this scheme, it may be necessary to buffer the burst in the optical burst switch until its control packet has been processed. In the JET scheme 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. The different signaling techniques for OBS networks is studied in detail in the next chapter.

Figure 1.3 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 avoiding their shortcomings.

1.2 Research Objectives

In this report, we analyze several critical issues affecting optical burst-switched networks, such as contention resolution, channel scheduling, burst assembly, signaling, and quality of service (QoS).

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 1.3. Comparison of the different all-optical network technologies.

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. We introduce a new approach, called *burst segmentation*, to reduce packet loss during contention resolution. Through simulation and analytical modeling, it is shown that segmentation policy reduces packet loss substantially when compared to the standard policy of dropping the contenting burst in the event of a contention. There are two ways of implementing burst segmentation with deflection namely, *Segment-First policy* and *Deflect-First policy*. In the Segment-First policy, the original burst is segmented and it's tail is deflected. While in the case of Deflect-First policy the contending burst is deflected if an alternate port is free, otherwise the original burst is segmented and its tail is dropped. We study the performance of both the policies with and without deflection and observe that policies with deflection outperform the standard dropping policy with and without deflection.

One of the key components in the design of optical burst-switched nodes is the development of efficient channel scheduling algorithms. In channel scheduling, multiple wavelengths are available on each link, and the problem is to assign an incoming burst to an appropriate channel or wavelength on the outgoing link. We propose non-preemptive and pre-

emptive scheduling algorithms that use burst segmentation to resolve data burst contentions during channel scheduling. We further reduce packet loss by combining burst segmentation and fiber delay lines (FDLs) to resolve contentions during channel scheduling. We propose two types of scheduling algorithms that are classified based on the placement of the FDL buffers in the optical burst-switched node. These algorithms are referred to as *delay-first* or *segment-first* algorithms. The simulation results show that the proposed algorithms can effectively reduce the packet loss probability compared to existing scheduling techniques.

QoS support is another important issue in OBS networks. Applications with diverse requirements urge transport technologies carrying the next-generation Optical Internet, such as OBS, to provide QoS guarantees. In this work, we propose three different approaches for handling prioritized data traffic. Our first approach of providing QoS support, is by introducing *prioritized burst segmentation* in the network core network. The prioritized burst segmentation scheme allows high-priority bursts to preempt low-priority bursts and enables full class isolation between bursts of different priorities. The proposed schemes are evaluated through analysis and simulation, and it is shown that prioritized burst segmentation provides 100% isolation between different classes of traffic, i.e., the performance of the high-priority traffic is not affected by the low-priority traffic. The approach can be easily extended to support multiple classes of traffic in a OBS networks.

In the second approach to providing QoS, we introduce a new technique for assembling packets into a burst, referred to as *composite burst assembly*. In this technique, a composite burst is created by combining packets of different classes into the same burst. The packets are placed from the head of the burst to the tail of the burst in order of decreasing class. The performance of this approach is enhanced by using a burst segmentation technique in which, during burst contention, only the packets in the tail of a burst are dropped. We describe a generalized model for burst assembly and burst scheduling, and we propose several composite burst assembly methods. The proposed schemes are evaluated through analysis and simulation, that having multiple class of packets in a burst performs better than having a

single class of packets in a burst.

In the third approach to providing QoS, we propose a *threshold-based burst assembly* scheme in conjunction with a burst segmentation policy to provide QoS in optical burst-switched networks. Bursts are assembled at the network edge by collecting packets that have the same QoS requirements. Once the number of packets in a burst reaches a threshold value, the burst is sent into the network. We investigate various burst assembly strategies which differentiate bursts by utilizing different threshold values or assigning different burst priorities to bursts that contain packets with differing QoS requirements. We show through simulation that there is an optimal value of burst threshold that in addition to providing QoS support, minimizes packet loss for given network parameters.

Signaling and reservation is one of the fundamental criteria upon which OBS can be differentiated from other all-optical transport technologies. OBS adopts an out-of-band signaling technique in which the burst header packet is sent ahead of the data burst by an offset time. The two commonly used signaling techniques in optical burst switching are tell-and-wait (TAW) and just-enough-time (JET). TAW suffers from high average end-to-end packet delay, while JET suffers from high packet loss probability. There is no signaling technique that offers flexibility in terms of both loss and delay tolerance.

We develop a generalized signaling framework for optical burst-switched networks. Based on the selection of the different parameters in the framework, we can understand the performance of the signaling technique. We also propose a hybrid signaling technique called *intermediate node initiated* (INI) signaling for optical burst-switched networks. INI can provide different levels of loss and delay characteristics based on the client applications requirements. Through simulation, we shown that INI performs better than TAW and JET in terms of average end-to-end packet delay and burst loss probability, respectively. We extend the INI signaling technique to provide differentiation in the core, by carefully choosing different initiation nodes based on the applications delay and loss requirements. We also show that the new signaling technique, *differentiated INI* (DINI) outperforms other existing QoS techniques.

In general, all the proposed solutions solve many of the fundamental issues faced by optical burst-switched networks, thereby making OBS more practical and deployable in the near future.

1.3 Organization of the Dissertation

This dissertation consists of nine chapters. This chapter has outlined a brief introduction to optical burst switching as well as the research objectives. Chapter 2 provides a survey of the current literature on the fundamental issues in optical burst switching, such as network architecture, burst assembly, routing and wavelength assignment, edge scheduling, signaling, channel scheduling, contention resolution, and quality of service. Chapter 3 proposes the concept of burst segmentation for contention resolution. Chapter 4 adopts the concept of burst segmentation, wavelength conversion, and optical buffering for scheduling arriving bursts on to outgoing data channels. Chapter 5 addresses the QoS issues while implementing prioritized burst segmentation with deflection routing. Chapter 6 proposes an another approach of providing QoS using composite burst assembly at the edge nodes. Both chapters 5 and 6 introduce many new QoS policies and also give the simulation results for each of the policies. An analytical loss model is also developed for both the prioritized burst segmentation and the composite burst assembly techniques. Chapter 7 provides QoS support in OBS networks using different threshold-based assembly policies. Chapter 8 describes a generalized signaling framework for OBS networks and proposes a new hybrid intermediate node-initiated (INI) signaling technique. INI is also extend to provide QoS using the differentiated intermediate node-initiated (DINI) signaling technique. Chapter 9 concludes the dissertation and identifies the possible areas of future work.

CHAPTER 2
OPTICAL BURST SWITCHING - A SURVEY

2.1 Introduction

Optical burst switching has been receiving attention as one of the most promising technologies to carry the next-generation optical Internet. Optical burst switching combines the advantages of optical circuit switching and optical packet switching, while overcoming their shortcomings. In OBS networks, the control plane and the data plane are separated, and the signaling is done out-of-band. Hence, the control plane can be electronic while the data plane is all-optical, making OBS practical to implement with current state of the art optical device technology.

The objective of this chapter is to provide a detailed survey of the OBS literature. We first start by understanding the architecture and then understanding the different functionality of each of the components in OBS. The following is the outline of this chapter. In Section 2.2 we describe the OBS network architecture, the edge node architecture, and the core node architecture. In Section 2.3, Section 2.4, and Section 2.5, we discuss the fundamental functionality of the edge nodes, such as burst assembly, routing and wavelength assignment, and edge scheduling. In Section 2.6, Section 2.7, and Section 2.8, we discuss the fundamental functionality of the core nodes, such as signaling, channel scheduling, and contention resolution. In Section 2.9, we provide an overview of all the existing techniques for providing QoS support in an OBS network.

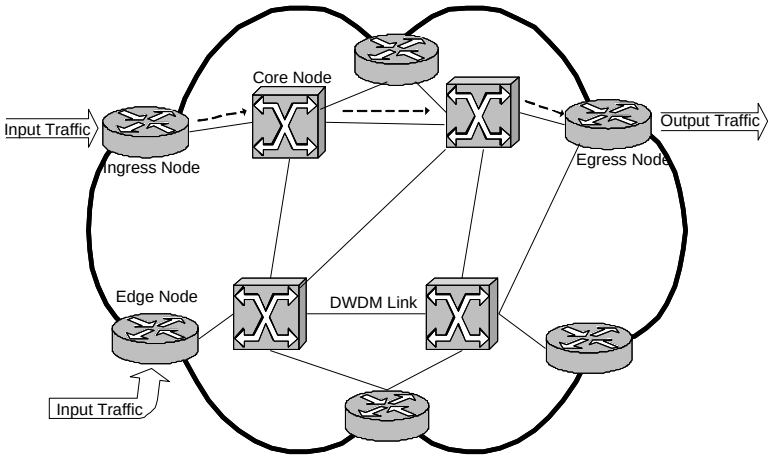


Figure 2.1. OBS Network Architecture

2.2 OBS Network Architecture

In optical burst switched networks, bursts of data consisting of multiple packets are switched through the network all-optically. A control message (or header) is transmitted ahead of the burst in order to configure the switches along the burst's route. The data burst follows the header without waiting for an acknowledgment that resources have been reserved and switches have been configured along the path.

Fig. 2.1 shows a OBS network. It consists of *edge nodes* and *core nodes*. An OBS network consists of optical burst switches interconnected with WDM links. An optical burst switch transfers a burst coming from an input port to its destination output port. Depending on the switch architecture, the node may or may not be equipped with optical buffering. The fiber links carry multiple wavelengths, and each wavelength can be seen as a channel. The control packet associated with a burst may also be transmitted in-band over the same channel as data, or on a separate control channel.

The edge routers assemble the electronic input packets into an optical burst, which is sent over the OBS core. The source edge router is referred to as the *ingress* node, and the destination edge router is referred to as the *egress* node. The ingress edge node assembles

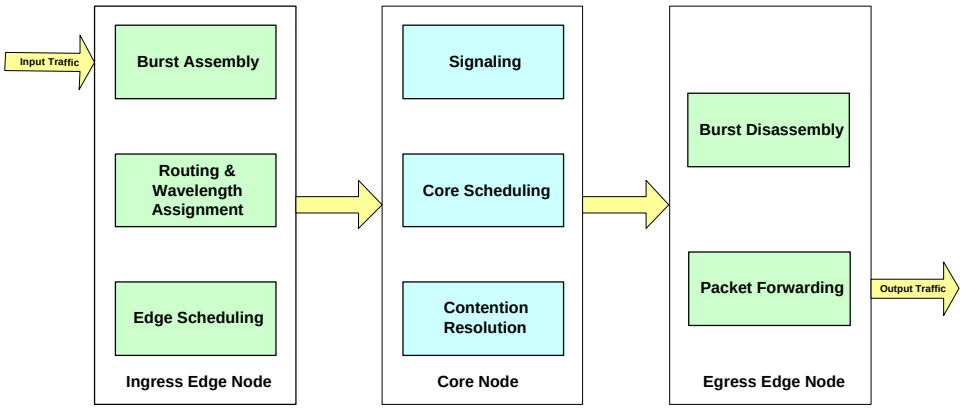


Figure 2.2. OBS functional diagram.

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 [28, 29, 30]. In this section, we will describe the edge and core node architecture, while the different functional components of an OBS network, as depicted in Figure 2.2, are described in the following sections.

In the network architecture, we assume 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(a) and Fig. 2.3(b).

The core routers (Fig. 2.3(a)) primarily 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 [31]. When the SCU receives a BHP, 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 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. 2.3(b)) performs the functions of pre-sorting packets, buffering packets, aggregating packets into burst, and de-aggregating 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 (BA), and a scheduler. The RM selects the appropriate output port for each packet and sends each packet to the corresponding BA module. Each BA module assembles bursts consisting of packets which are headed for a specific egress router. In the BA 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) [32]. 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 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.

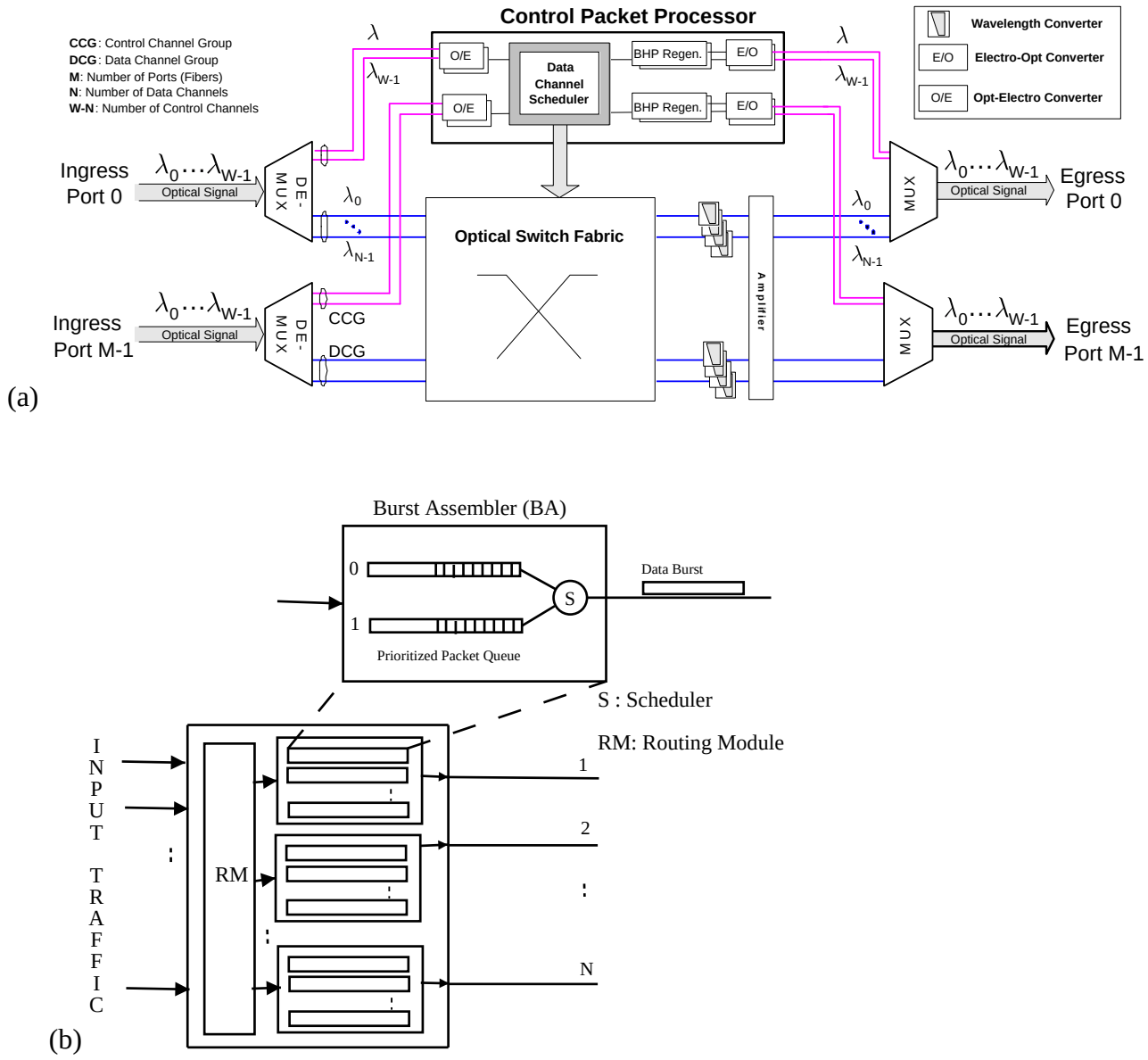


Figure 2.3. (a) Architecture of Core Router. (b) Architecture of Edge Router.

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. The authors of WR-OBS claim that this design improves the network throughput; but the centralized nature of the design is not very scalable.

In the following section, we discuss the edge node issues of burst assembly, routing and wavelength assignment, and edge scheduling.

2.3 Burst Assembly

Burst assembly is the process of aggregating and assembling input packets from the higher layer into bursts at the ingress edge node of the OBS network. The trigger criterion for the creation of a burst is very important, since it predominantly 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 [33]. 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 processing time at each node and the ratio of the control channels to the number of data channels in the fiber [28].

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 2.4 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 [34], 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) [35].

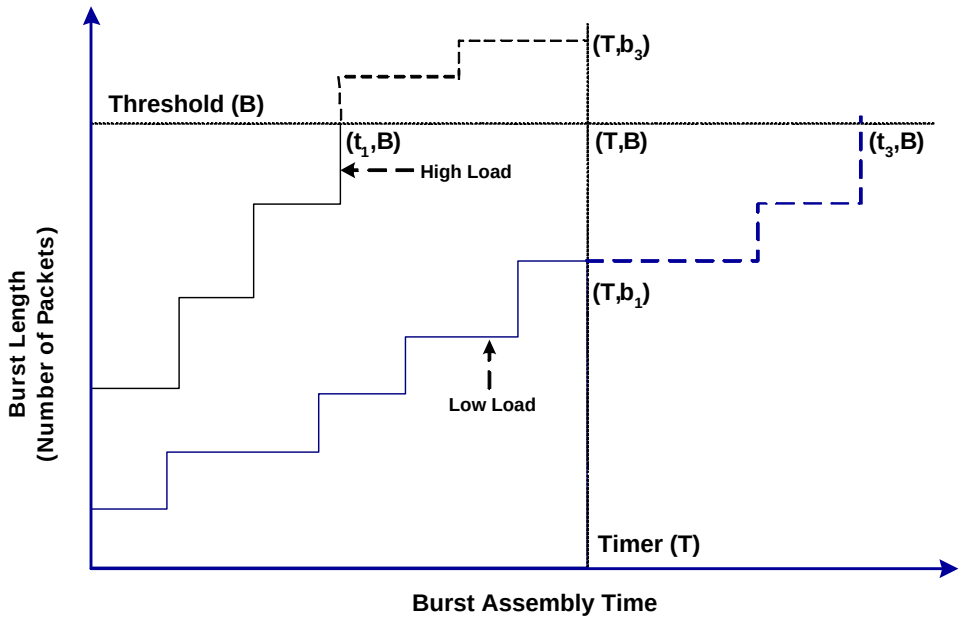


Figure 2.4. Effect of load on timer-based and threshold-based aggregation techniques.

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.

In [36], 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 [19], the signaling scheme needs to know both the arrival time and the length of the burst in advance. While in JIT [37, 38], 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 [39], 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.

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 [40, 41, 42] 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 [43], 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 [40, 41, 42], 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 [40, 41] show that the correlation structure at large to infinite time scales still does not change.

To the best of our knowledge, there is no existing work which investigates the optimal burst length for minimizing packet losses in the optical core. In Chapter 7, we present some results indicating the presence of an optimal burst length in a network given specific load vales.

During burst assembly, the ingress node pre-sorts and schedules the incoming packets into electronic input buffers according to each packet's QoS class and destination address. The packets are then aggregated into bursts that are stored in the output buffer. Since a separate packet buffer is required for each packet class and each destination, the limit on the maximum number of supported packet class is determined by the maximum electronic packet buffer size and the maximum number of packet queues at each ingress node.

A more complicated situation arises when packets arriving to an ingress node are of different classes. In this case, packets must be assembled into bursts, and priorities assigned to bursts in a manner which will enable the optical core to provide different levels of service to each class of packets. The choice of a single burst assembly mechanism for all classes of traffic may be inappropriate. A threshold-based scheme or a timer-based scheme with a

high timer value may lead to unacceptable delays for packet classes with delay constraints, while non-optimal burst lengths may lead to higher loss for packets with loss constraints. In Chapter 6, we investigate a novel approach to burst assembly, referred to as *composite burst assembly*. In composite burst assembly, packet of different classes with different QoS requirements may be assembled into a single burst. We have found that composite burst assembly techniques can provide different levels of service for packets of different classes within the same burst if appropriate contention resolution mechanisms are implemented within the optical core. In Chapter 7, we investigate a *differentiated burst assembly* technique for supporting different classes of traffic. In differentiated burst assembly, burst types are defined based on packet QoS requirements. Each burst type is then assembled using an appropriate assembly mechanism to ensure QoS requirements. The timer value is decided based on the end-to-end delay constraint of the constituent packets, and the threshold values is set to the optimal burst length for a given load range.

2.4 Routing and Wavelength Assignment

Routing is one of the fundamental aspects of any transport technology. In the current literature, most of the OBS researchers assume fixed source routing. The hop-by-hop routing followed in IP networks is not suitable due to the long per-hop route computation duration. Also, most of the literature assumes that a fixed shortest path is calculated at the source to the destination. The shortest path can be based on either the shortest physical distance path in the case of a strict delay constraint, or the minimum hop path in the case of a strict loss constraint.

Multi-protocol label switching (MPLS) proposed by the IETF can be adopted for routing in OBS networks. In MPLS-based OBS routing, each burst header packet (BHP) is assigned a *label* based on the mapped forward equivalent class (FEC) at the source edge node. The intermediate nodes switch the burst based on the assigned labels to the intended destination [21, 22, 38, 44]. Such a label-switched optical burst-switching technique is referred to as *labeled optical burst switching* (LOBS).

Fixed source routing goes hand in hand with MPLS, since it is possible to pre-assign a single label along the entire switched path. The disadvantage of a labeled switched approach is during link or node failures. All the traffic on the path will be lost unless the concerned node at the failure point is updated with the new path information that is calculated around the point of failure. In order to handle a link or a node failure, several *protection* and *restoration* techniques have been extensively studied for wavelength-routed optical networks [45, 46, 47, 48, 49, 50, 51, 52, 53]. Recently, a few solutions for handling link failure in an OBS network have been proposed. In [54], the standard *1+1 protection* scheme is adopted for handling failure in an OBS network. In [55, 56], the authors evaluate fast protection and restoration techniques for OBS networks. [57, 58] investigate several rerouting algorithms using ILPs for providing load-balanced link-disjoint alternate paths for each specific link failure in a Labeled OBS network.

In optical burst-switched networks, data loss may occur when bursts contend for network resources. There have been several proposed solutions to resolve contentions in order to minimize data loss (details can be found in Section 2.8). These localized contention resolution techniques react to contention, but do not address the more fundamental problem of congestion. Hence, there is a need for network-level contention avoidance using load-balanced routing techniques in order to minimize data loss. In [59], two dynamic congestion-based load balanced routing techniques are proposed to avoid congestion. The *Congestion-Based Static-Route Calculation* technique pre-computes two link-disjoint route for each source-destination pair and dynamically selects one of the routes based on the current congestion along the two paths. In the *Least-Congested Dynamic Route Calculation* technique, a least-congested route is calculated dynamically at periodic intervals. The simulation results show that the proposed contention avoidance techniques improve the network utilization and reduce the data loss. In [60, 61], the authors investigated a similar load-balancing routing approach using adaptive alternate path routing and concluded with similar observations as [59].

The assignment of a specific wavelength for each arriving burst can be done in one of two approaches. In the first approach, the burst is assigned the same wavelength along the entire path from the source to its destination, imposing the wavelength-continuity constraint along the entire path. One of the primary disadvantages of this approach is the increased burst loss due to the unavailability of the chosen wavelength through the entire path. Such an approach is particularly useful if the OBS network does not support all-optical wavelength converters at the core nodes or in the case of the WR-OBS network [32], in which the centralized request scheduler performs both the routing and wavelength assignment.

In the second approach, the wavelength-continuity constraint is not enforced with the assumption that each OBS node has wavelength conversion capability (all-optical wavelength converter). In this approach, each OBS node can assign a different outgoing wavelength to every arriving burst based on the set of available wavelengths. Most of the existing literature assumes all-optical wavelength converter at each OBS node [21, 10, 28]. We discuss in-detail the issue of selecting an outgoing wavelength for an arriving burst in Sections 2.5 and 2.7.

An important concern during routing and wavelength assignment in OBS networks is the fairness of loss experienced by data transmission on longer versus shorter paths. There has been some effort to provide fairness based on different path lengths.

The offset time of a burst reduces as the burst travels through the network, leading to higher blocking probability at the end of long routes. In [62], an algorithm ranks bursts based on how many network resources they have already consumed and how close they are to their destinations, and uses this ranking to implement a preemptive priority regime. Through simulation, the authors conclude that this algorithm produces substantial reduction in the blocking probability of the OBS network.

In [63], the authors suggest increasing the offset time between the BHP and the data burst at every hop as the burst travels from the source to its destination. By increasing the offset time at every hop, the burst has a higher probability of being successfully reserved at the

next hop along the path. The side-effect of this scheme is additional per-hop delay penalty incurred in the burst transmission. The paper [64] also presents an approach to provide fairness based on providing additional offset in OBS networks. The authors present a link scheduling state-based offset selection (LSOS) scheme to solve the fairness problem.

The authors of [65] propose a parallel reservation technique in which bursts of longer hop-length paths can be scheduled on any of the available wavelengths at an OBS node, while bursts of shorter hop-length paths can be scheduled on a limited subset of wavelengths.

2.5 Edge Scheduling

Once a burst is assembled, it must be scheduled for transmission over the optical core in a manner which satisfies the QoS requirements of the packets contained in the burst. This problem is referred to as the *edge scheduling* problem. In most of the existing literature on optical burst switching, a burst is assumed to be transmitted as soon as the burst is assembled at the ingress node [66, 43]. However, in an optical burst-switched network, it is possible that the output port may be occupied by another burst originating at the same ingress node, or the output port may be occupied by an optical transit burst originating at a different ingress node. In this case, transmitting a new burst when the output port is busy would result in a contention. In edge scheduling, once a burst is created, it is placed in an output burst queue corresponding to the burst's priority and desired output port. The output port can either be occupied or available, and there may be multiple bursts waiting for the same output port. The problem is first to decide whether or not to preempt the currently serviced burst, if any, which is occupying the output port. If the current burst is preempted or if the output port is idle, the problem is then to select one of the waiting bursts to transmit.

Edge scheduling can be viewed as the problem of sending the created bursts into the core such that the loss, delay, and bandwidth constraints of each class are met. Edge scheduling is similar in some respects to traditional packet scheduling in IP routers and switches. In IP networks, packets are transported in a store-and-forward manner, with packets being

sorted into prioritized buffers at each node, waiting to be scheduled for transmission. Similar to IP networks, in an optical burst-switched network, the created bursts will be sorted at the ingress node according to output port. However, in IP networks, each output port is normally associated with a static point-to-point transmission link. Hence, in the case of a contention at the source, where the intended output port is occupied by a transit burst of Priority P_x, the edge scheduling policy has to take into account the relative priorities of each new burst versus P_x. To guarantee QoS of packet classes, the mapping between burst priorities and burst types is an important issue in OBS networks.

The IP QoS literature is rich with packet scheduling policies [67]. It may be possible to adapt these policies for optical burst-switched networks. We assume that once bursts are created, they are placed in a prioritized burst queue corresponding to the appropriate output port. Following are possible burst scheduling approaches derived from existing IP scheduling approaches.

- First-Come-First-Served (FCFS): Bursts are served in the same order that they are created.
- Priority queuing (PQ): Each prioritized burst queue is a FCFS queue. A burst is scheduled to an output port only if all burst queues of higher priority are empty.
- Round Robin (RR): Burst are assigned a priority and are placed in separate prioritized burst queue. One burst from each queue is sent into the network in a round robin fashion [68].
- Weighted Round Robin (WRR): Each prioritized burst queue is served in a round-robin order. In each round, the number of bursts sent depends on the weight assigned by the policy. The weight may be assigned based on bandwidth requirements or based on the burst priority. This is similar to Class Based Queuing (CBQ).
- Waiting Time Priority (WTP): Bursts are assigned a priority and are placed in separate prioritized burst queues. The priority of a burst is initially calculated based on the QoS

requirements of the burst's packets, and the priority increases with waiting time. The scheduler chooses the burst at the head of the queue with the highest priority, and sends this burst into the core.

- Class Based Queuing (CBQ): This approach is a variation of priority queuing. CBQ is based on the notion of controlled link sharing, and is designed to avoid resource denial to a particular class of service. Bursts are assigned a priority and placed in separate prioritized burst queues. We can define the service preference for each of the queues and the amount of queued traffic in bytes to be transmitted from each queue on each service rotation. For each service rotation, the appropriate amount of traffic is transmitted from each queue. In the case of empty queues, the other queues can "borrow" the unused bandwidth based on the "borrow" attribute value for that queue. Hence, CBQ provides a graceful method of preempting the prioritized queues, thereby avoiding resource denial and resource starvation.

Most of the above discussed edge scheduling algorithms do not consider the status of the core network. In [69], a pro-active scheduling algorithm referred to as burst overlap reduction algorithm (BORA) is proposed. The idea behind BORA is based on the observation that if the total number of simultaneously arriving bursts at an output port exceeds the number of channels at that port, burst loss will be inevitable. Thus, if we can reduce the total number of simultaneously arriving bursts from a given source at each port, it is likely that the burst loss will be reduced. BORA tries to pro-actively avoid burst contention at remote (downstream) nodes. The basic idea is to serialize the bursts on outgoing links to reduce the burst overlapping degree (and thus burst contention and burst loss at downstream nodes). This can be accomplished by judiciously delaying locally assembled bursts beyond the pre-determined offset time using the electronic memory available at the ingress nodes. The results show that the loss rate of BORA is much lower than the loss in existing algorithms. The biggest side-effect of BORA is that it introduces significant delay during the serialization of the bursts.

So far we have discussed all the functionality performed at the OBS edge nodes. In the following sections we will discuss the functionality of the core nodes, such as signaling, channel scheduling, and contention resolution.

2.6 Signaling

Signaling and reservation is one of the fundamental criteria upon which OBS can be differentiated from other all-optical transport technologies. OBS adopts an out-of-band signaling technique in which the burst header packet is sent ahead of the data burst by an offset time.

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 at each node. The signaling may either be implemented *in-band*, in which a control message, or a burst header, is transmitted on the same wavelength as the data burst, or *out-of-band*, in which the burst header is transmitted on a separate wavelength from the data burst.

After the header is transmitted, the source node may either wait for an acknowledgment before transmitting the data burst, or the source node may transmit the data burst without first receiving an acknowledgment. The method in which the source node waits for an acknowledgment is referred to as *tell-and-wait* (TAW). In TAW the resources are guaranteed to be reserved; however, the end-to-end delay may be higher due to the additional time spent waiting for an acknowledgment.

When the source node does not wait for an acknowledgment, the data burst will either follow the header immediately, a signaling method referred to as *tell-and-go* (TAG), or the data burst will follow the header after some offset time (see Fig. 2.7), a signaling method referred to as *just-enough-time* (JET).

In this section, we will develop a generalized signaling framework and also describe the individual OBS signaling protocols.

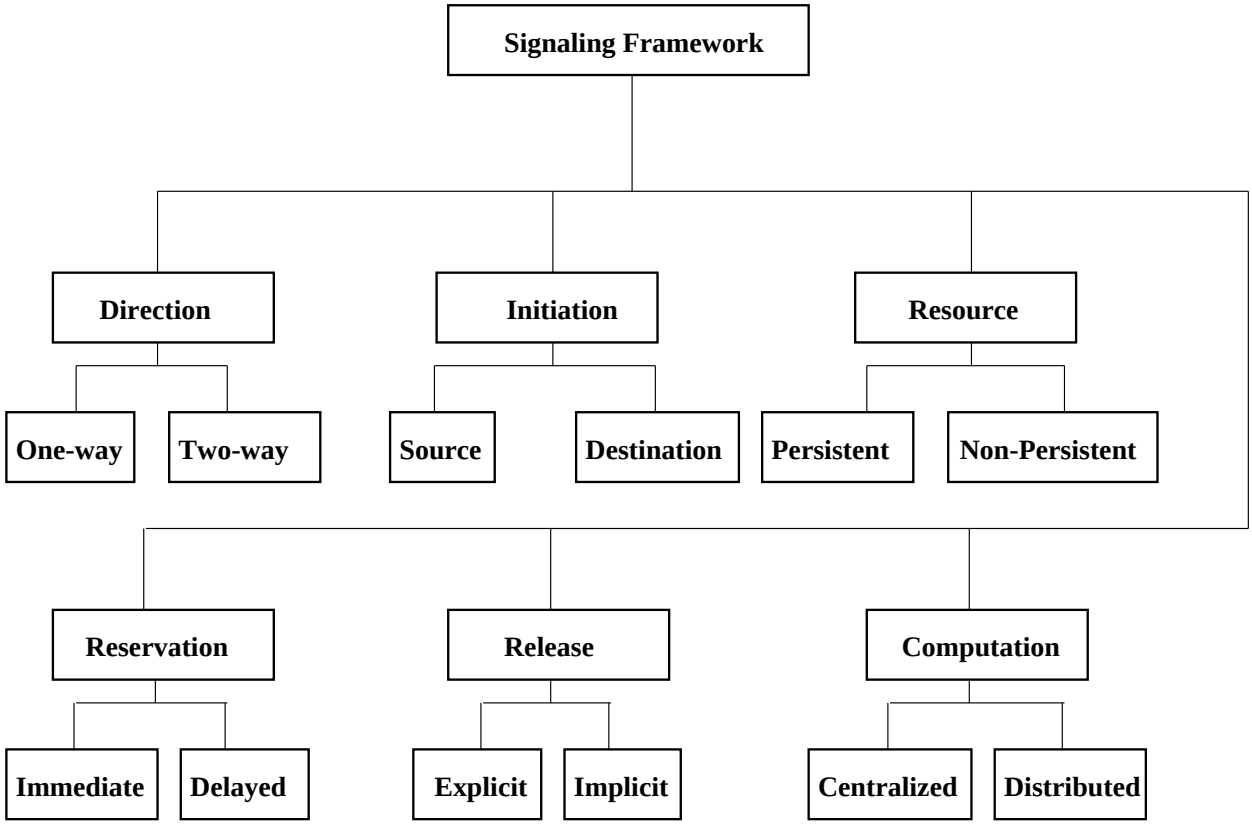


Figure 2.5. Generalized signaling framework.

2.6.1 Generalized Signaling Framework

Signaling is a critical aspect that can significantly affect the performance of a network. For OBS networks, signaling is even more important, since the core is (usually) bufferless and any contention for resources during signaling can lead to data loss. In this section, we aim to develop a generalized signaling framework, which can aid in the careful evaluation of all design parameters before opting for a particular signaling technique, given the requirements of the data to be transmitted. We first look at the different design parameters that affect the performance of a signaling technique.

- *One-way or Two-way:* The connection setup phase of any signaling technique can be either one-way or two-way. In one-way based signaling, the source sends out a control packet requesting the intermediate nodes in path to allocate the necessary resources for

the data burst. No acknowledgment message is sent back to the source notifying the source of the success or failure of the resource reservation. The primary objective of the one-way based signaling techniques is to minimizes the end-to-end data transfer latency (delay). Unfortunately, this objective leads to high data loss due to contention of data bursts.

Two-way based signaling techniques are acknowledgment-based, where the request for a resource is sent from the source to the destination. The acknowledgment message confirming a successful assignment of requested resources is sent back from the destination to the source. The data burst is transmitted only after a connection is established successfully. If any of the intermediate nodes in the path are busy, then the request is blocked. That particular intermediate node takes suitable actions to release all the previously reserved links (if any), and also transmits a failure message back to the source. The source can choose to retry or drop the request. The primary objective of the two-way based technique is to minimize packet loss, but such an objective leads to high data transfer delay due to the round-trip delay during connection setup.

- *Source Initiated or Destination Initiated Reservation:* A signaling technique can initiate reserving the requested resources at the source or at the destination. In the *source initiated reservation (SIR)* technique, the resources are reserved in the forward path from the source to the destination. If the resource allocation is successful in the forward direction, an acknowledgment message containing the reserved wavelength is sent back to the source. The source, upon receiving the resource confirmation, transmits the burst into the core network. 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), and send a reservation request back to the source node, through the intermediate nodes, to reserve the chosen wavelength. 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 [70, 71].

- *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.

- *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 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

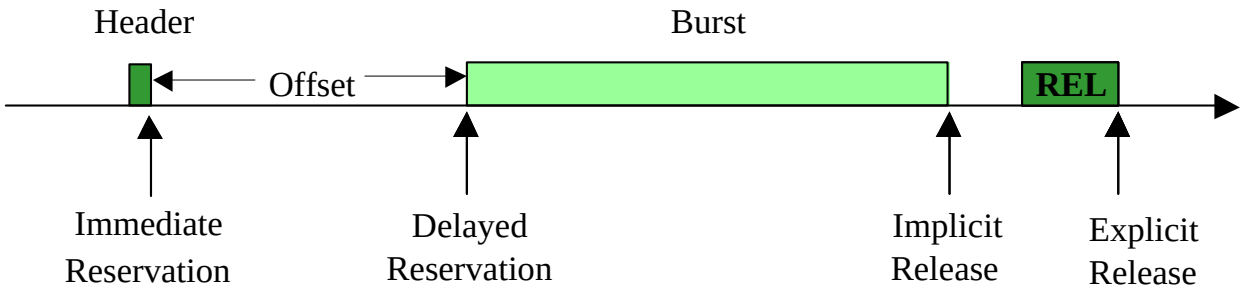


Figure 2.6. Reservation and Release Mechanisms.

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.

- *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. 2.6), the signaling techniques can be categorized into four categories, *Immediate Reservation / Explicit Release* , *Immediate Reservation / Implicit Release*, *Delayed Reservation / Explicit Release* , and *Delayed Reservation / Implicit Release* [38, 72]. 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 reser-

vation, 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.

- *Centralized or Distributed:* In a *centralized signaling* technique, as proposed by [32], 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. 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 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.

2.6.2 Just-Enough-Time (JET)

Figure 2.7 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

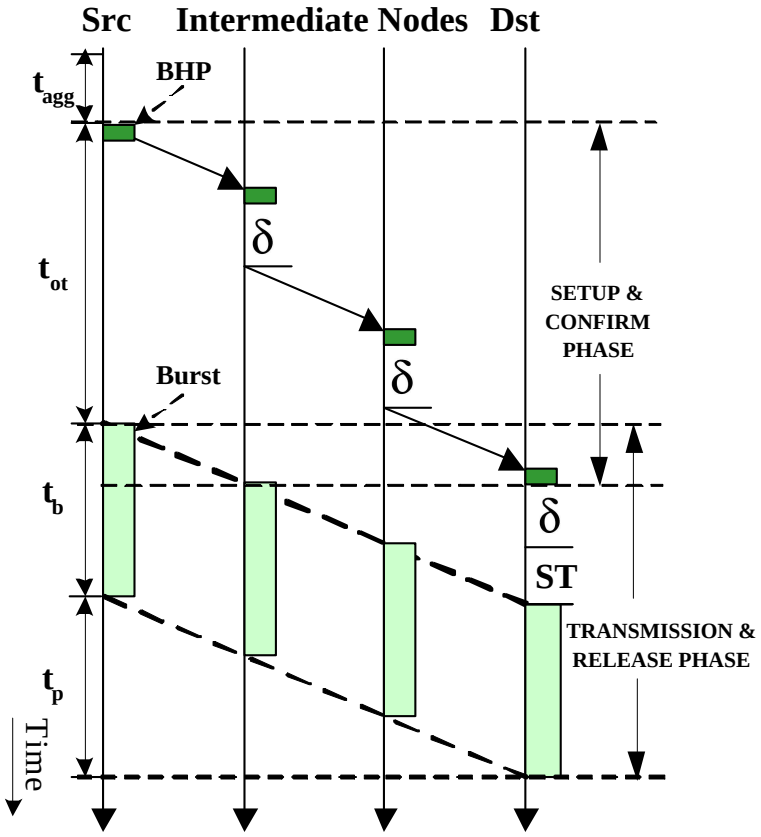


Figure 2.7. Just-Enough-Time (JET) signaling technique.

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 [10]. The offset time is calculated based on the number of hops from source to destination, and the switching 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

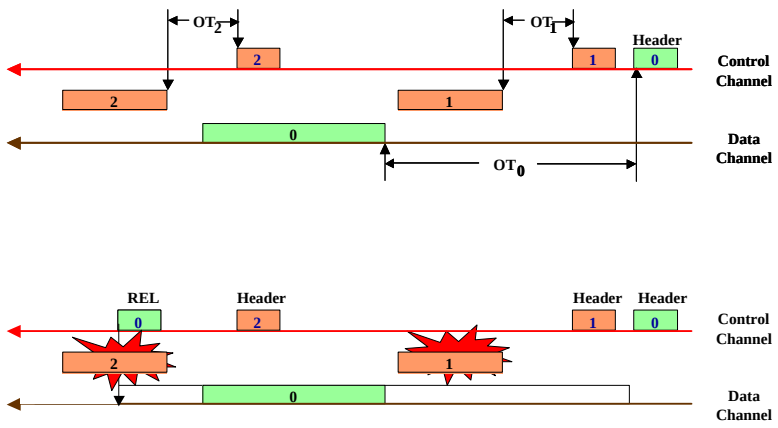


Figure 2.8. Comparison of (a) JET and (b) JIT based signaling.

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 approach, 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. 2.8(a) and (b) compares a similar signaling scenario using JET and JIT, respectively. An architectural framework for implementing various JIT schemes is presented in [38]. 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.

2.6.3 Tell-and-Wait (TAW)

Figure 2.9 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)* [71, 70]. 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.

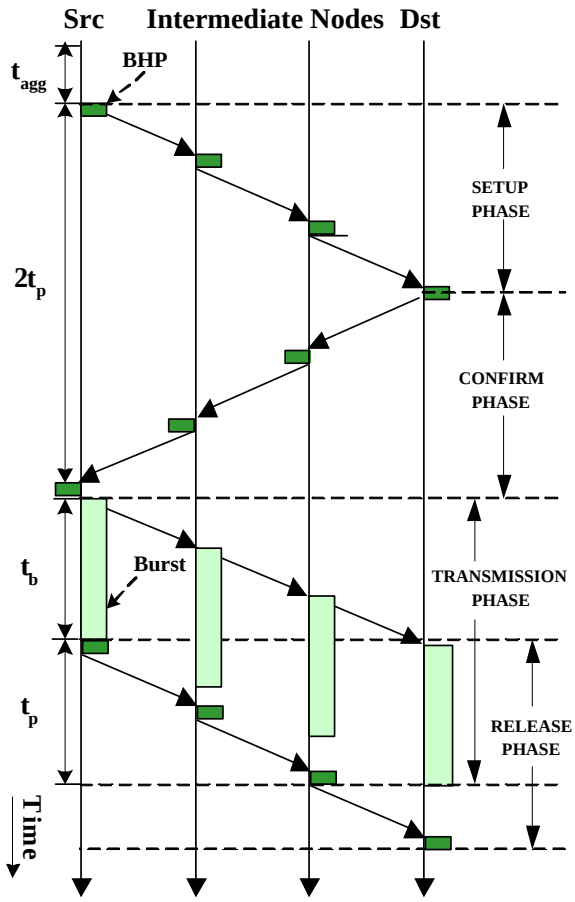


Figure 2.9. Tell-and-Wait (TAW) signaling technique.

2.7 Channel Scheduling

Another type of scheduling in optical burst-switched networks is *channel scheduling*. In channel scheduling, multiple wavelengths are available on each link, and the problem is to assign an incoming burst to an appropriate channel or wavelength on the 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. 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. 2.10(a) and (b), respectively. In Fig. 2.10(a), the $LAUT_i$ on every data channel D_i, $i = 0, 1, ..., W$, is maintained by the

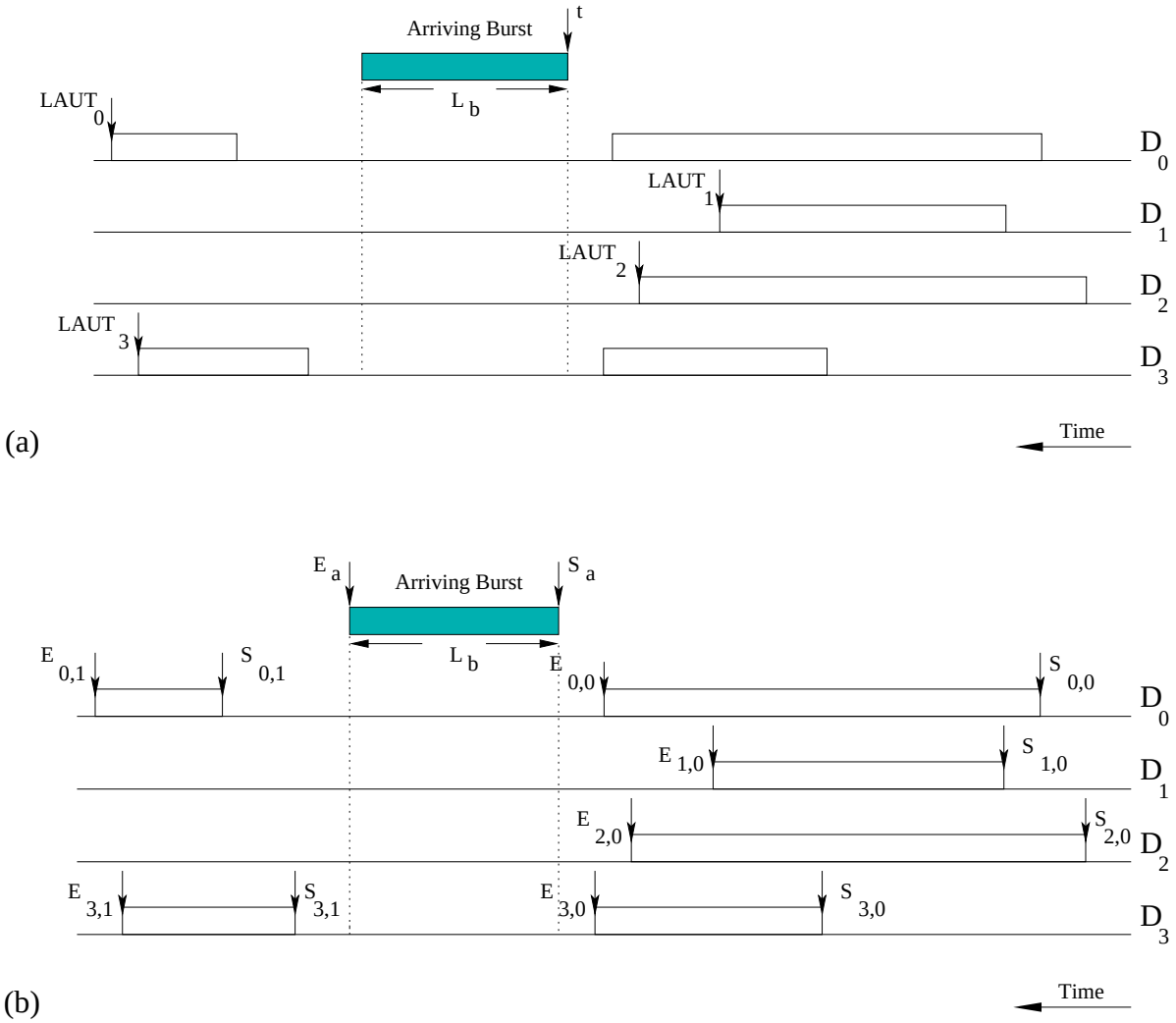


Figure 2.10. Initial data channel status (a) without void filling (b) with void filling.

scheduler. In Fig. 2.10(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 Filling (FFUC-VF) and Latest Available Unscheduled Channel with Void Filling (LAUC-VF).

2.7.1 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. 2.11(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 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.

2.7.2 Horizon or Latest Available Unscheduled Channel (LAUC):

The LAUC or Horizon [21] 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. 2.10(a). Based on the $LAUT_i$, data chan-

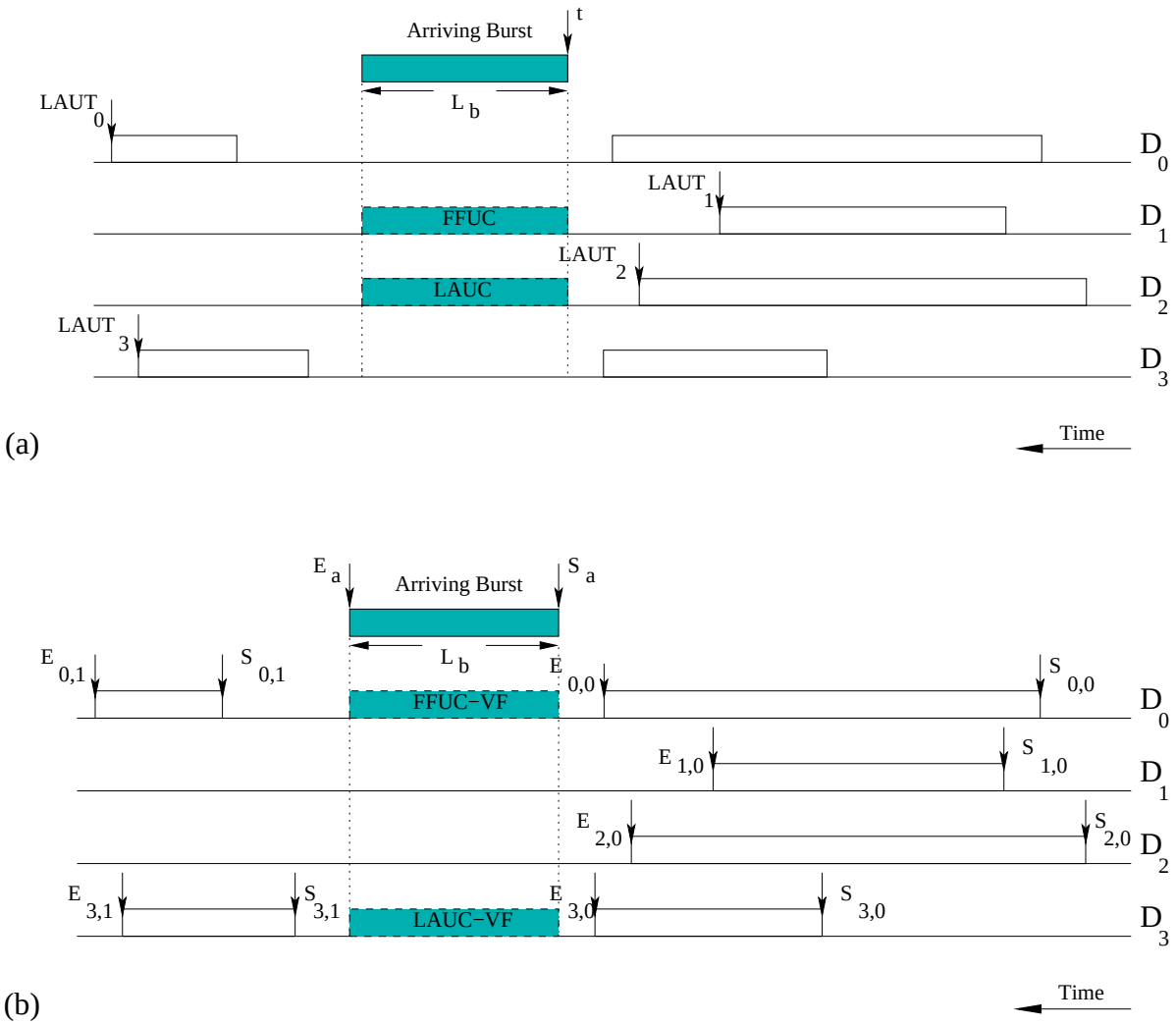


Figure 2.11. Channel assignment after using (a) non void filling algorithms (FFUC and LAUC), and (b) void filling algorithms (FFUC-VF and LAUC-VF).

nels 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 $O(\log W)$.

2.7.3 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. 2.10(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 $O(\log(WN_b))$.

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

The LAUC-VF [73] 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. 2.10(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 [74], 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.

[75] 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 cri-

teria 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) [76] or a Group-based Scheduling algorithm [77], 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 [28, 21, 78]. In Chapter 6.4, we described several scheduling algorithms that are based on burst segmentation [79], with and without FDLs. We shown that our proposed algorithms can achieve significantly lower loss than all the above scheduling algorithms [80, 81].

2.8 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. This is a problem that commonly arises in packet switches, and is known as *external blocking*. External blocking is typically resolved by buffering all but one of the contending bursts. switch, techniques designed to address the contention (external blocking) problem include optical buffering, wavelength conversion, and deflection routing. Whether these approaches will prove adequate to address the external blocking problem is still an open issue. Below we look at each of these solutions.

2.8.1 Optical Buffering

Typically, 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)* [82, 83, 84, 85, 86] can be utilized to delay packets for a fixed amount of time. By implementing multiple delay lines in stages [84] or in parallel [85], 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 [87, 88]. In [87], the buffer size is increased by cascading multiple stages of delay lines. In [88], 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 [89, 90].

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 [82, 91]. 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 [82, 92, 7] or multistage feed-forward shared buffering [83, 84, 86]. Furthermore, buffers can be either configured as *degenerate buffer* (linear increment) or *non-degenerate buffer* (non-linear increment) [93, 88].

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.

2.8.2 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 [94].

In optical burst switching with wavelength conversion, contention is reduced by utilizing additional capacity in the form of multiple wavelengths per link [95, 21]. 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 [96]. 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 restricting 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 [97, 6, 98].

2.8.3 Deflection Routing

Deflection routing or hot-potato routing is ideally suited to switches that have little buffer space. This approach of resolving contention is to route the contending bursts to an output port other than the intended output port [99, 100, 101]. However, the deflected burst may end up following a longer path to its destination. As a result, the end-to-end delay for a burst may be unacceptably high. Also, packets will have to be re-ordered at the destination since they are likely to arrive out of sequence. 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 all-optical burst-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 all-optical burst-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.

While deflection routing has been investigated for electronic and photonic packet-switched networks [99, 100, 101], there is currently little work which applies deflection to optical burst-switched networks.

In [99], hot-potato routing is compared to store-and-forward routing in a ShuffleNet. [100] and [101] 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 [102], 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 [103], 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 [104, 92, 105].

The deflected bursts may end up following a longer path to the destination, leading to higher end-to-end delay, and packets may also arrive at the destination out-of-order [99, 101, 100]. A combination of contention resolution techniques may be used to provide high throughput, low delay, and low packet loss probability.

In deflection routing, a deflected 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 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 [103, 104].

In JET-based optical burst-switched networks, another concern when implementing deflection is the offset time. As the burst traverses each hops, the offset between the burst and its header decreases; thus, it is possible that, if the burst traverses a large number of hops, the burst may overtake the header. Approaches for ensuring the sufficient separation of the header and data at each node include setting a higher initial offset value at the source node and using FDLs at each intermediate node to delay the burst.

An approach to further reduce packet loss due to contention called *burst segmentation* [106] is proposed in the next chapter. Burst segmentation is the process of dropping only those parts of a burst which overlap with another burst. A variation of segmentation in which overlapping segments of the head of the latter arriving burst are dropped is described in [107].

2.9 Quality of Service

QoS support is another important issue in OBS networks. Applications with diverse requirements urge transport technologies carrying the next-generation Optical Internet, such as OBS, to provide QoS guarantees.

There are two 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 relatively in comparison 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, such as multimedia and mission-critical applications. Moreover, from the ISP'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 fixed ahead of time and satisfied using different techniques.

Most QoS differentiation schemes in IP networks focus on providing loss differentiation, delay differentiation, or bandwidth guarantees, since IP routers have the capability to buffer packets electronically. However, OBS core nodes do not have any electronic buffers, and the bursts follow an all-optical path from source to destination. Thus, the delay incurred

from source to destination is primarily due to propagation delay, and bandwidth guarantee is implicitly provided by supporting loss guarantee. Hence, the focus of QoS support in OBS networks is to primarily provide loss differentiation, though there are a few papers addressing the problem of providing delay differentiation.

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 the packets are scheduled onto an output port only if all packet queues of higher priority are empty. Weighted fair queuing [108] 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 [109] and [110] 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 [111] 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 [112], 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.

In OBS networks, several schemes have been proposed to support the relative QoS model. A differentiated signaling scheme may be used to provide QoS in optical burst-switched networks. In [23, 95], 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 this additional offset-based reservation scheme, higher-priority class bursts are given a larger offset time than the lower-priority class bursts. By providing a

larger offset time, the probability of reserving the resources without conflict for the higher-priority class burst is increased, and therefore, the loss experienced by higher-priority class traffic is decreased. The limitation of this approach is that high-priority bursts will experience higher delays; thus, the approach may be capable of satisfying loss requirements, but is not capable of meeting delay requirements. Furthermore, it has been shown that this scheme can lead to unfairness, with larger low-priority bursts experiencing higher loss than smaller low-priority bursts [66, 113].

Contention resolution schemes may be used to provide QoS in an all-optical core network. In [66], 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,

$$\frac{p_i}{p_j} = \frac{s_i}{s_j}. \tag{2.1}$$

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.

In [114], another QoS approach based on priority queueing was proposed for OBS networks. The scheme incorporates the LAUC-VF (Section 2.7.4) 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.

In [115], 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.

Service differentiation is also provided by different burst assembly schemes. In [66], 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 [28], the packets are classified 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 [116, 35]. In [28], 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 [117], the authors address the problem of providing QoS support by implementing a differentiated Look-ahead window Contention Resolution (LCR) algorithm. Simulation results show that the look-ahead contention resolution algorithm can readily support service differentiation and offers high overall performance with moderate complexity. The authors claim that the LCR algorithm can be modified to reduce the total end-to-end burst delay at the cost of slightly lowering the performance.

In [118], 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).

The authors in [119, 120, 121, 122], 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.

In [123], the authors have proposed QoS schemes based on the physical quality of the optical signal, such as signal-to-noise ratio (SNR), maximum 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.

In [124], 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.

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 [5] 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, we need to develop efficient absolute QoS differentiation schemes in which all wavelengths in the network are available for statistical multiplexing and dynamic bandwidth allocation.

In [125, 126], the authors propose an absolute QoS model that provides a worst-case loss probability for the guaranteed traffic. Two mechanisms for providing loss guarantees at OBS core nodes are an early dropping mechanism, which probabilistically drops the non-guaranteed traffic, and a wavelength grouping mechanism, which provisions necessary wavelengths for the guaranteed traffic are proposed. It is shown that integrating these two mechanisms outperforms other schemes in providing loss guarantees, as well as reducing the loss experienced by the non-guaranteed traffic. The authors also discuss admission control and resource provisioning for OBS networks, and propose a path clustering technique to further improve the network-wide loss performance [127]. Analytical loss models for the proposed schemes are developed and verified by the simulation results.

In [128], 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 [129] 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.

We have recently introduced a scheme in [130] for optical burst-switched networks. The scheme utilizes deflection as well as a contention resolution approach referred to as *burst segmentation* (described in Chapter 3) to resolve contentions. Our 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 is completely isolated from the low-priority bursts contentions.

CHAPTER 3

BURST SEGMENTATION: AN APPROACH FOR REDUCING PACKET LOSS IN OPTICAL BURST-SWITCHED NETWORKS

3.1 Introduction

A major concern in optical burst-switched networks is contention, which occurs when multiple bursts contend for the same outgoing link at a given node. Contention in an optical burst-switched network is particularly aggravated by the variable burst sizes and the long burst durations. Furthermore, since bursts are switched in a cut-through mode rather than a store-and-forward mode, optical burst-switched networks generally do not have very much buffering capabilities.

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 [84, 85]. Current optical buffer architectures are severely limited in size; thus, nodes in an all-optical network may be unable to handle high loads or bursty traffic without alternative contention resolution schemes. With wavelength conversion, contention is reduced by utilizing additional capacity in the form of multiple wavelengths per link [99, 101, 100]. A contending burst may be switched to any of the available wavelengths on the outgoing link. 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 existing contention resolution schemes, such as optical buffering, wavelength conversion, and deflec-

tion routing, may be utilized in optical burst-switched networks, additional schemes may still be necessary in order to further reduce high contention rates and to reduce loss and achieve higher network utilization.

In the current literature, most approaches to contention resolution address the minimization of burst losses rather than packet losses. In existing contention resolution schemes for optical burst-switched networks, when a 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, i.e., the *contention period*, may be minimal. For certain applications which have stringent delay requirements but relaxed packet loss requirements, it may be preferable to lose a few packets from a given burst rather than losing the entire burst. In this chapter, we will introduce a new contention resolution technique called *burst segmentation*, in which only those packets of a given burst which overlap with another burst will be dropped.

The remainder of this chapter is organized as follows. Section 3.2 introduces the concept of burst segmentation and describes the segment dropping policies. Section 3.3 discusses segmentation with deflection. Section 3.4 describes an analytical loss model for the burst segmentation technique. Section 3.5 compares the analytical and simulation results for different contention resolution policies in a specific network topology, and Section 3.6 concludes the chapter.

3.2 Burst Segmentation

To overcome some of the limitations of optical burst switching, we introduce the concept of burst segmentation. 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. 3.1). 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. 3.2. If the switching time is non-negligible, then additional segments may be lost when the switch is being reconfigured.

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,

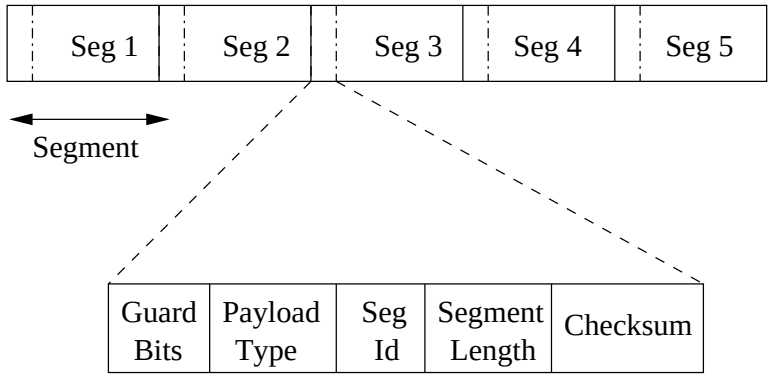


Figure 3.1. Segments header details.

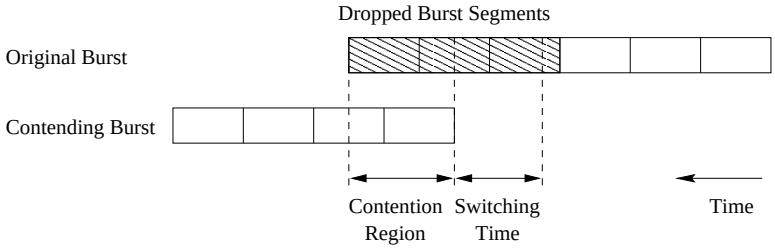


Figure 3.2. Selective segment dropping for two contending bursts.

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.

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. 3.2) 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.

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. 3.3 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. 3.4, 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.

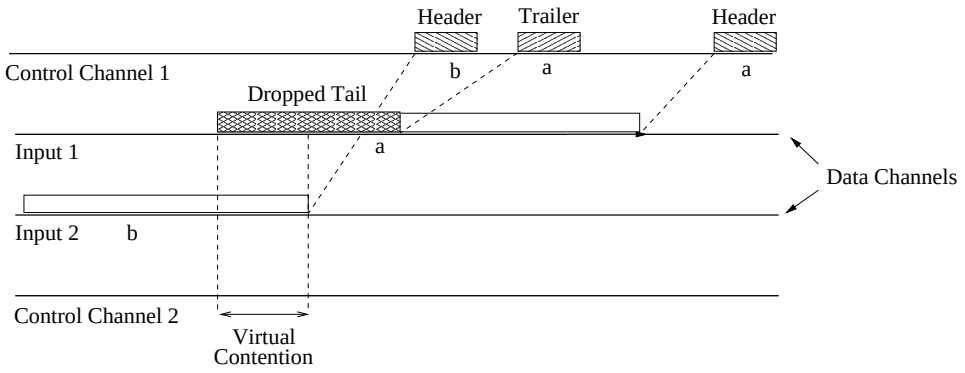


Figure 3.3. Trailer packet effective.

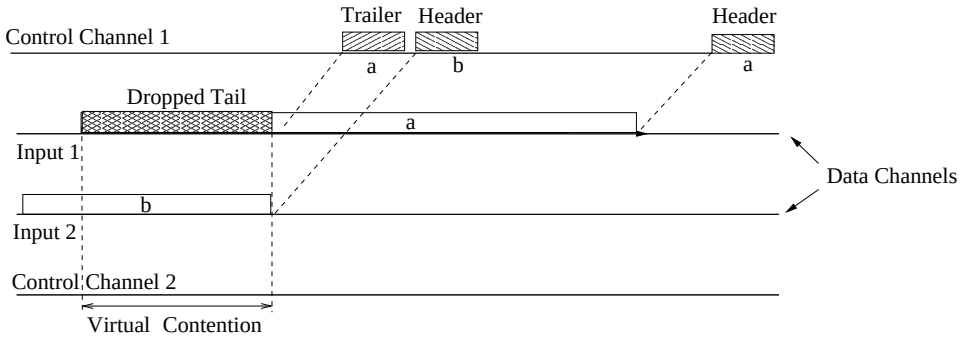


Figure 3.4. Trailer packet ineffective.

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 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) [131, 132] 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 [133], but have yet to be deployed in practical

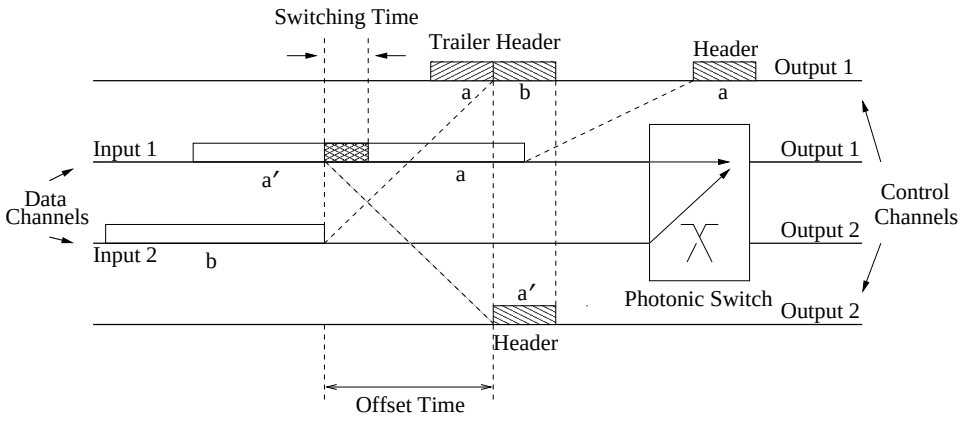


Figure 3.5. Segmentation with deflection policy for two contending bursts.

systems. In our experiments, we assume an intermediate and more practical switching time of 10 microseconds.

3.3 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 [99, 101, 100]. Implementing segmentation with deflection (Fig. 3.5) 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 [103]. 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 [134, 135, 136].

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. 3.5). The second approach is to incorporate priorities into the burst. In this case, the lower-priority burst is deflected or segmented [137].

When combining segmentation with deflection, there are two 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 original burst is dropped.

An example of the segmentation-deflection scheme is shown in Fig. 3.5. 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 chapters 5,6, and 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.

3.4 Analytical Loss Model

In this section, we develop an analytical model for evaluating the packet loss probabilities with burst segmentation in which no length comparison is done (SDP). We assume that bursts arrive to the network according to a Poisson process with rate λ^{sd} bursts per second for source-destination pair sd respectively. Fixed routing is assumed, and no buffering is supported at

core nodes. We also assume that all bursts have the same offset time. This implies that the BHP of the original burst always arrives before the BHP of the contending burst. Traffic on each link is assumed to be independent. Also, the switching time is assumed to be negligible. We begin by defining the following notation:

- λ_l^{sd}: arrival rate of bursts to link l, on the path between source s and destination d.
- $\lambda_l = \sum_{sd} \lambda_l^{sd}$: arrival rate of bursts to link l, due to all source-destination pairs sd.
- r_{sd}: route from source s to destination d.

The load placed on a link l by traffic going from source s to destination d depends on whether link l is on the path to destination d. If link l is on the path to d, then the load applied to link l by sd traffic is simply λ^{sd}. Thus,

$$\lambda_l^{sd} = \begin{cases} \lambda^{sd} & \text{if } l \in r_{sd} \\ 0 & \text{if } l \notin r_{sd}. \end{cases} \tag{3.1}$$

Also, the total (new) burst arrival into the network, λ, is given by:

$$\lambda = \sum_s \sum_d \lambda^{sd}. \tag{3.2}$$

We calculate the packet loss probability by finding the distribution of the burst length at the destination and comparing the mean burst length at the destination to the mean burst length at the source. Let the initial cumulative distribution function of the burst length be $G_{l_0^{sd}}(t)$ for bursts transmitted from source s to destination d, where l_0^{sd} is the zeroth hop link between source s to destination d. The cumulative distribution function of the burst after k hops is $G_{l_k^{sd}}(t)$. Let $F_{l_k^{sd}}(t)$ be the cumulative distribution function for the arrival time of the next burst on the k^{th} hop link l between source-destination pair sd:

$$F_{l_k^{sd}}(t) = \quad 1 - e^{-\lambda_{l_k^{sd}} t}, \tag{3.3}$$

where $\lambda_{l_k^{sd}}$ is the arrival rate of all bursts on the k^{th} hop link of the path between source s and destination d.

The burst length will be reduced if another burst arrives while the original burst is being transmitted; thus, the probability that the burst length is less than or equal to t after the first hop is equal to the probability that the initial burst length is less than or equal to t or that the next burst arrives in time less than or equal to t. Therefore,

$$\begin{aligned} G_{l_1^{sd}}(t) &= 1 - (1 - G_{l_0^{sd}}(t))(1 - F_{l_1^{sd}}(t)) \\ &= 1 - (1 - G_{l_0^{sd}}(t))e^{-\lambda_{l_1^{sd}} t}. \end{aligned} \tag{3.4}$$

Similarly, let $G_2(t)$ be the cumulative distribution function of the burst after the second hop:

$$\begin{aligned} G_{l_2^{sd}}(t) &= 1 - (1 - G_{l_1^{sd}}(t))(1 - F_{l_2^{sd}}(t)) \\ &= 1 - (1 - G_{l_0^{sd}}(t))e^{-(\lambda_{l_1^{sd}} + \lambda_{l_2^{sd}})t}. \end{aligned} \tag{3.5}$$

In general,

$$\begin{aligned} G_{l_k^{sd}}(t) &= 1 - (1 - G_{l_{k-1}^{sd}}(t))(e^{-\lambda_{l_k^{sd}}}) \\ &= 1 - (1 - G_{l_0^{sd}}(t))e^{-\left(\sum_{i=1}^{k} \lambda_{l_i^{sd}}\right)t}. \end{aligned} \tag{3.6}$$

We now find the expected length after k hops and compare this length with the expected length at the source node in order to obtain the expected loss that a particular burst will experience. Let $L_{l_k^{sd}}$ be the expected length of the burst at the k^{th} hop.

Case (1): If we have fixed-sized bursts of length, $\frac{1}{\mu} = T$, then the initial distribution of the burst length is given by:

$$G_{l_0^{sd}}(t) = Pr(T \leq t) = \begin{cases} 1 \text{ if } t \geq T \\ 0 \text{ if } t < T. \end{cases} \tag{3.7}$$

Substituting (3.7) into (3.6) and taking the expected value, we obtain:

$$L_{l_k^{sd}} = \frac{1 - e^{-\sum_{i=1}^{k} \lambda_{l_i^{sd}} T}}{\sum_{i=1}^{k} \lambda_{l_i^{sd}}}. \tag{3.8}$$

Case (2): If the initial burst length is exponentially distributed, we have:

$$G_{l_0^{sd}}(t) = 1 - e^{-\mu t}. \tag{3.9}$$

Substituting (3.9) into (3.6) and taking the expected value, we obtain:

$$L_{l_k^{sd}} = \frac{1}{\sum_{i=1}^{k} \lambda_{l_i^{sd}} + \mu}. \tag{3.10}$$

We now find the expected length after K hops, where K is the total number of hops between s and d. Let $Loss_{sd}$ be the expected length of the burst lost per burst for a burst traveling from s to d:

$$Loss_{sd} = \frac{1}{\mu} - L_{l_K^{sd}}. \tag{3.11}$$

Hence, the packet loss is proportional to the length of the route and the length of the burst. The packet loss probability of bursts, P_{loss}^{sd}, is then given by:

$$\begin{aligned} P_{loss}^{sd} &= \frac{E[Length\ Lost]}{E[Initial\ Length]} \\ &= Loss_{sd} \cdot \mu. \end{aligned} \tag{3.12}$$

We can then find the average packet loss probability of bursts for the system by finding the individual loss probability for each source-destination pair, and taking the weighted average of the loss probabilities:

$$P_{loss} = \sum_{s} \sum_{d} \frac{\lambda^{sd}}{\lambda} P_{loss}^{sd}. \tag{3.13}$$

3.5 Numerical Results

In order to evaluate the performance of the proposed schemes and to verify the analytical models, a simulation model is developed. Fig. 3.6 shows the 14-node NSF network on which the simulation and analytical results are applied. The link distances are shown in km.

3.5.1 Analytical Results

In the analytical model, we ignore the switching time and header processing time at each intermediate node along the path of the burst.

Figure 3.7 plots the packet loss probability versus load for the segment drop policy (SDP) with exponential burst length and fixed-sized bursts. In SDP, the contending burst al-

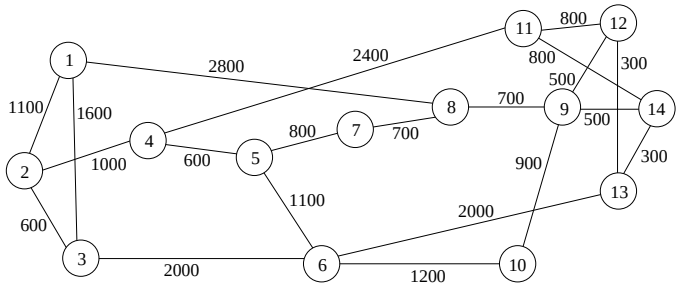


Figure 3.6. Picture of NSFNET with 14 nodes (distance in km).

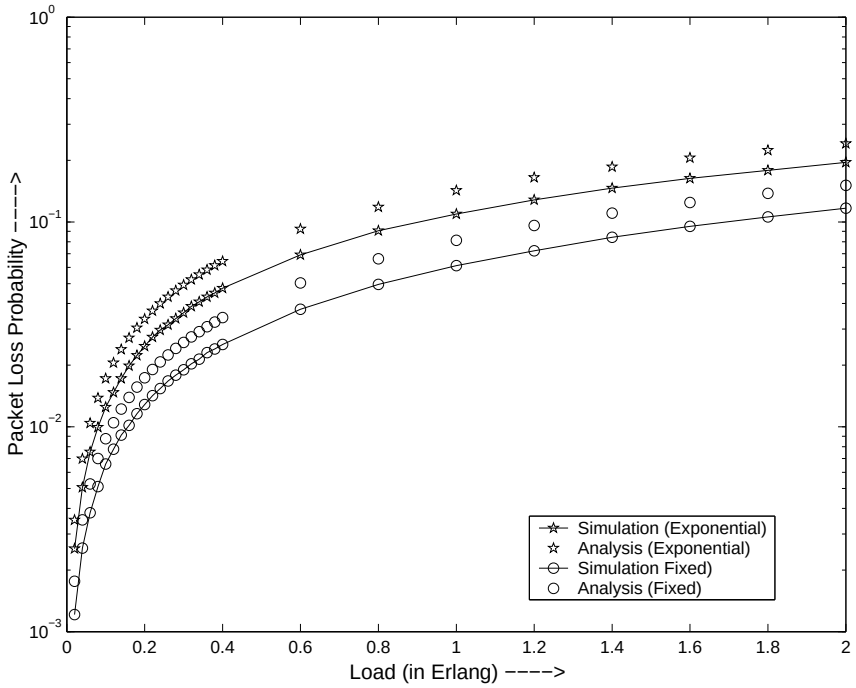


Figure 3.7. Packet loss probability versus load for both exponential initial burst size, $1/\mu$ = 100 ms and fixed initial burst size = 100 packets, using SDP without length comparison.

ways preempts the original burst. We observe that the analytical model slightly over-estimates the packet loss probabilities due to the independent link assumption. We also observe that the packet loss with fixed-sized bursts is lower than packet loss with exponentially distributed burst sizes, since the maximum number of packets lost per contention is potentially less with a fixed initial burst size. This observation may be useful when determining the burst assembly policy (studied in Chapter 7).

3.5.2 Simulation Results

In order to evaluate the performance of the segmentation and deflection schemes, we develop a simulation model. 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.
- Traffic is uniformly distributed over all source-destination pairs.
- Fixed shortest path routing is used between all node pairs.

Figure 3.8 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

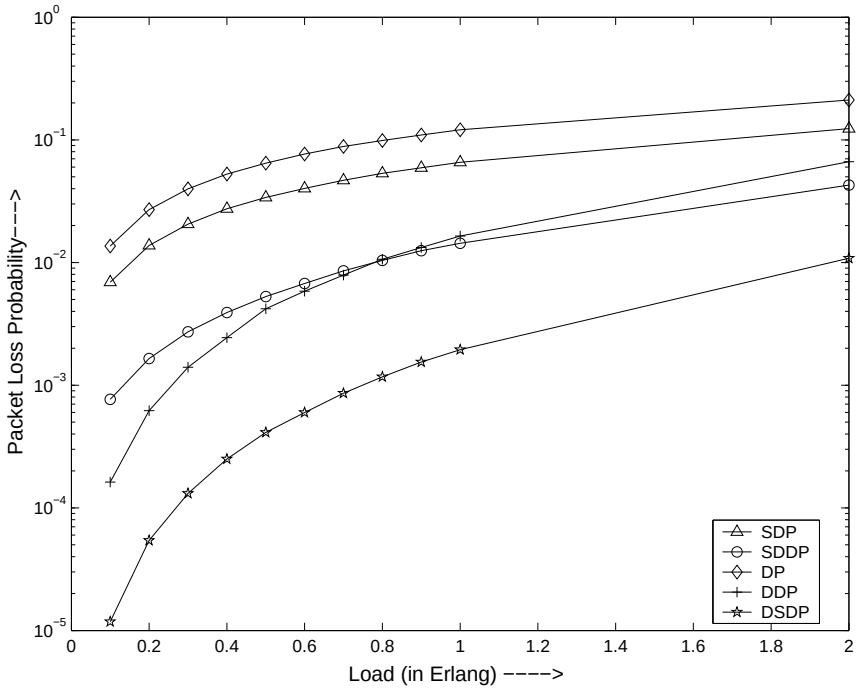


Figure 3.8. Packet loss probability versus load for NSFNET at low loads with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

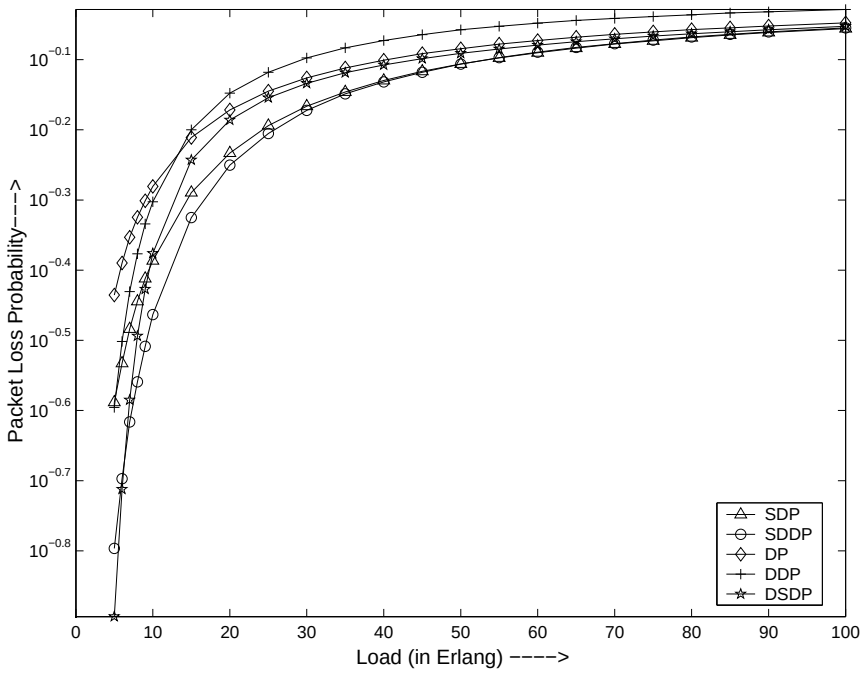


Figure 3.9. Packet loss probability versus load for NSFNET at high loads with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

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 3.9 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 3.10 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 3.11 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

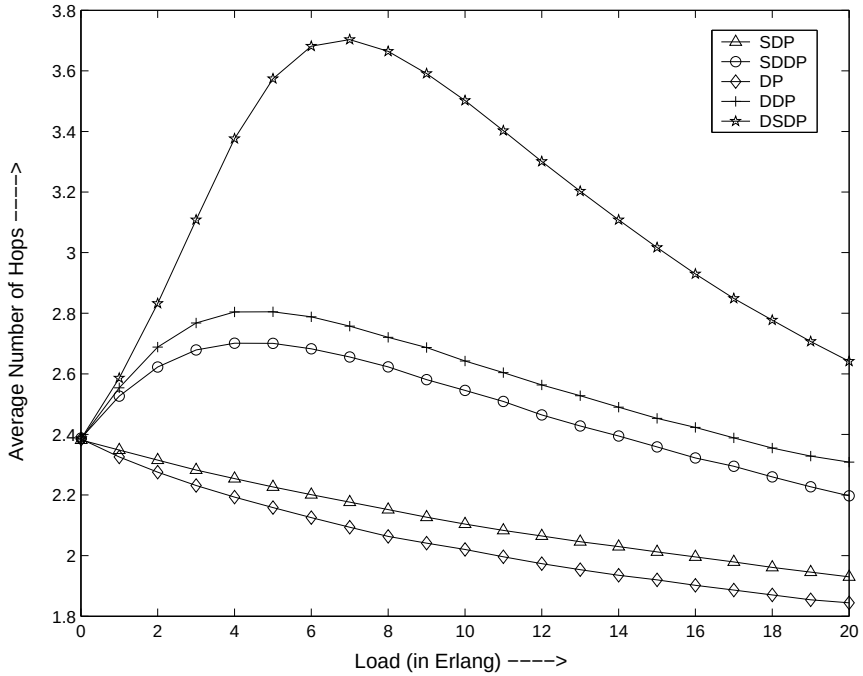


Figure 3.10. Average number of hops versus load for NSFNET with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

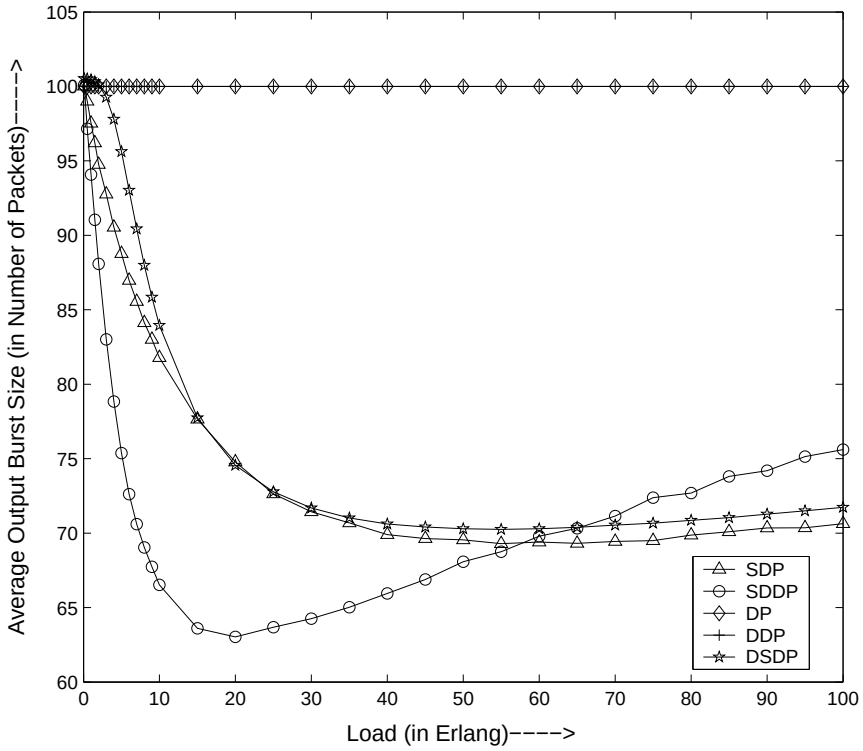


Figure 3.11. Average output burst size versus load for NSFNET with $\frac{1}{\mu} = 100$ μs and Poisson burst arrivals.

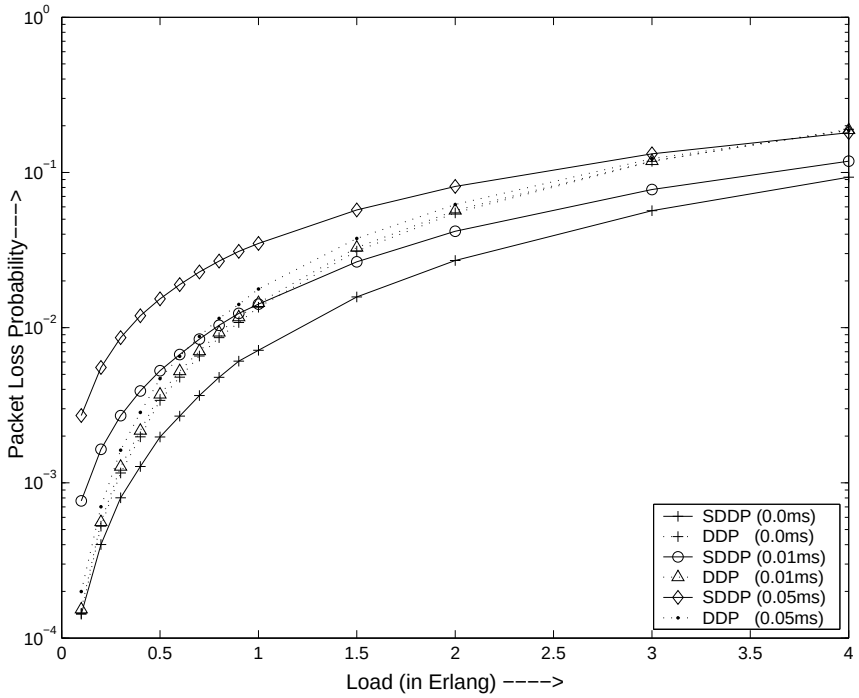


Figure 3.12. Packet loss probability versus load at varying switching times for NSFNET with $\frac{1}{\mu} = 100\mu$s and Poisson burst arrivals.

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. 3.12. 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.

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,

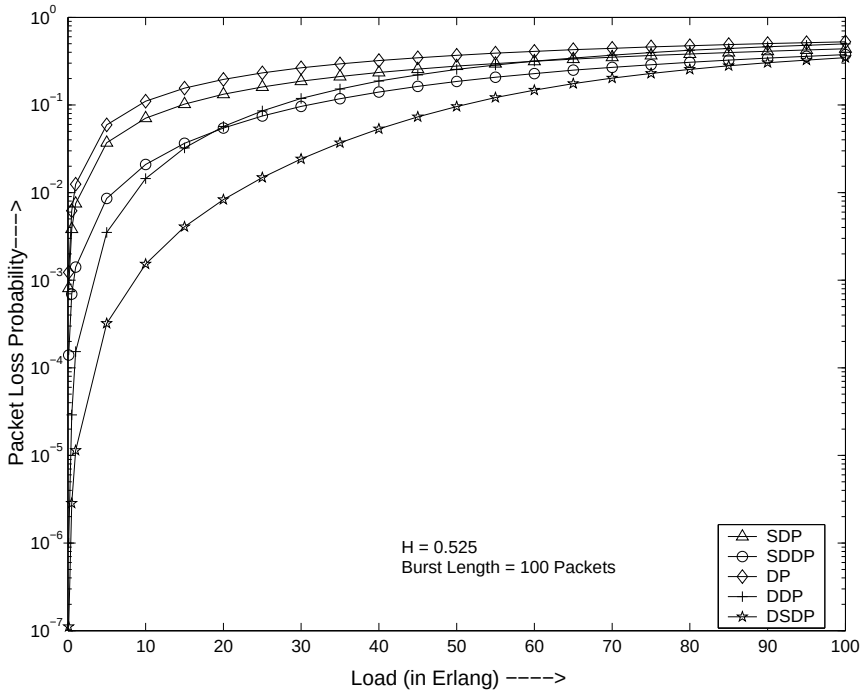


Figure 3.13. Packet loss probability versus load for NSFNET with Pareto burst arrivals.

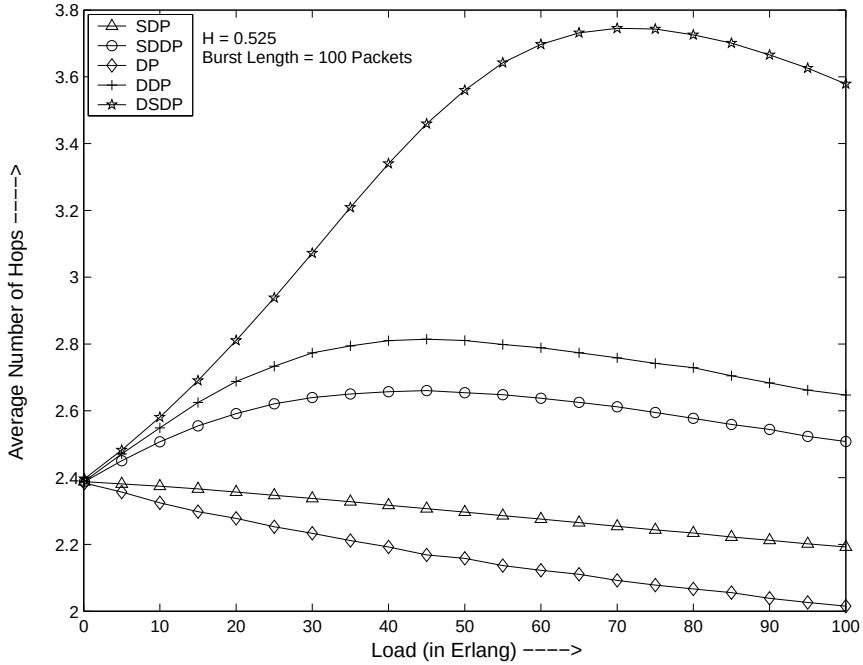


Figure 3.14. Average number of hops versus load for NSFNET with Pareto burst arrivals.

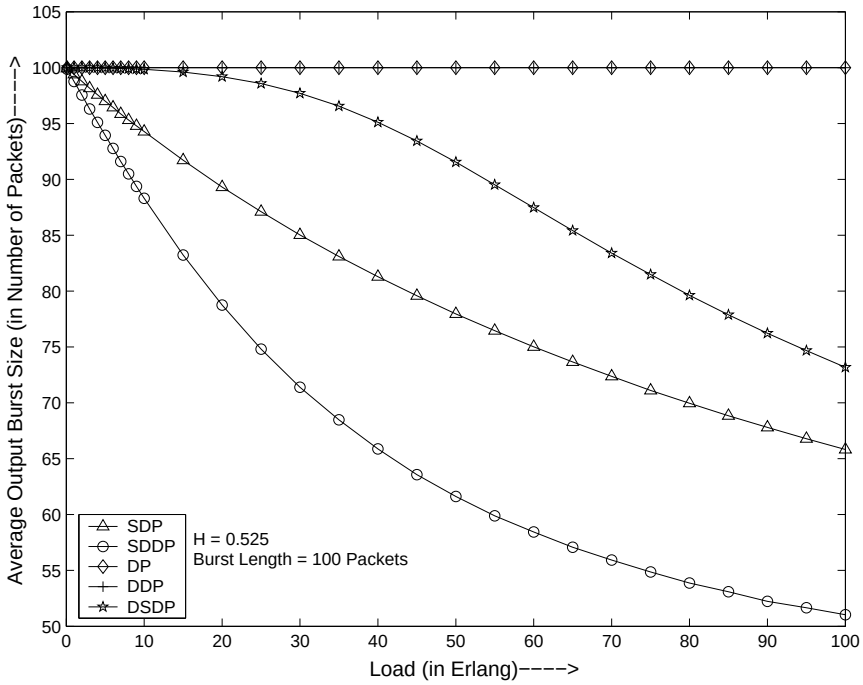


Figure 3.15. Average output burst size versus load for NSFNET with Pareto burst arrivals.

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 3.13 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 3.14 shows the average number of hops versus load with Pareto burst arrivals for the policies. Figure 3.15 shows the average output burst size versus load with Pareto burst arrivals, for the different policies. The results are similar to the Poisson case.

3.6 Conclusion

In this chapter, we proposed burst segmentation as a new contention resolution technique for optical burst-switched networks. Segmentation provides packet-level (or segment-level) loss granularity in a burst-switched network. Segmentation provides the lower-bound in packet loss due to burst contention in an OBS network. Segmentation can also work well in conjunction with all other contention resolution techniques, such as optical buffering, wavelength conversion, and deflection routing. We also investigated a number of different policies with and without segmentation and deflection. The segmentation policies perform better than the standard dropping policy, and offer the best performance at high loads. The policies which incorporate deflection tend to perform better at low loads. We also developed an analytical loss model for burst segmentation and the analytical results were verified by matching simulation results.

A number of other works been published acknowledging the benefit of our proposed segmentation technique as a new contention resolution technique for optical burst-switched networks. In [107, 138, 139], the head-dropping segmentation scheme is evaluated. The authors of [140, 141, 142, 143] also evaluate the benefits of segmentation in an OBS network.

The segment drop policy (SDP), discussed in this chapter, introduces the possibility of preemption in an all-optical burst-switched network. Preemption can be used to provide differentiated services in an all-optical core for different traffic classes based on applications with different QoS requirements. We investigate several segmentation-based QoS techniques in Chapters 5, 6, and 7.

CHAPTER 4

SEGMENTATION-BASED CHANNEL SCHEDULING ALGORITHMS FOR OPTICAL BURST-SWITCHED NETWORKS

4.1 Introduction

One of the primary reasons for data loss in OBS networks is burst contentions. When two or more bursts are destined for the same output port at the same time, a contention occurs. There are many contention resolution schemes [103] which may be used to resolve the contention. The primary contention resolution schemes are optical buffering, wavelength conversion, deflection routing, and burst segmentation. In optical buffering, fiber delay lines (FDLs) are used to delay the burst for a specified amount of time, proportional to the length of the delay line, in order to avoid the contention [84]. In wavelength conversion, if two bursts on the same wavelength are destined to go out of the same port at the same time, then one burst can be shifted to a different wavelength [94]. In deflection routing, one of the two bursts will be routed to the correct output port (primary) and the other to any available alternate output port (secondary).

In burst segmentation [106], the burst is divided into basic transport units called *segments*. Each of these segments may consist of a single IP packet or multiple IP packets, with each segment defining the possible partitioning points of a burst when the burst experiences contention in the optical network. All segments in a burst are initially transmitted as a single burst unit. However, when contention occurs, only those segments of a given burst that overlap with segments of another burst will be dropped, as shown in Fig. 4.1. If switching time is not negligible, then additional segments may be lost when the output port is switched from one burst to another.

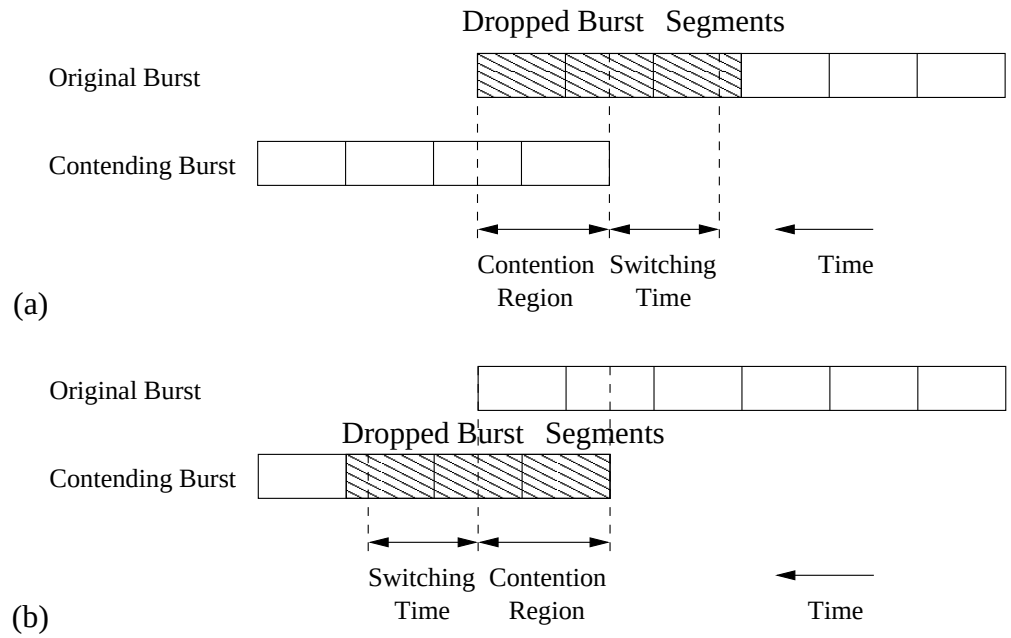


Figure 4.1. Selective segment dropping for two contending bursts (a) tail dropping policy (b) head dropping policy.

There are two approaches for dropping burst segments when contention occurs between bursts. The first approach, tail dropping, is to drop the tail of the original burst (Fig. 4.1(a)), and the second approach, head dropping, is to drop the head of the contending burst (Fig. 4.1(b)) [106].

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)[21]. Void filling algorithms include first fit unscheduled channel with void filling (FFUC-VF) and latest available unscheduled channel with void filling (LAUC-VF) [28]. The performance of scheduling algorithms can be enhanced by using optical buffering (FDLs), wavelength converters, and deflection routing techniques for resolving burst contentions [21, 28, 78, 144, 145, 146]. 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

[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*.

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 (Chapter 5).

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 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 [28], 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 this chapter, we study new 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. The rest of the chapter is organized as follows. In Section 4.2, we discuss the OBS core node architecture with the scheduler and also describe two core node architectures with FDLs. Section 4.3 describes the proposed data channel scheduling algorithms with and without void

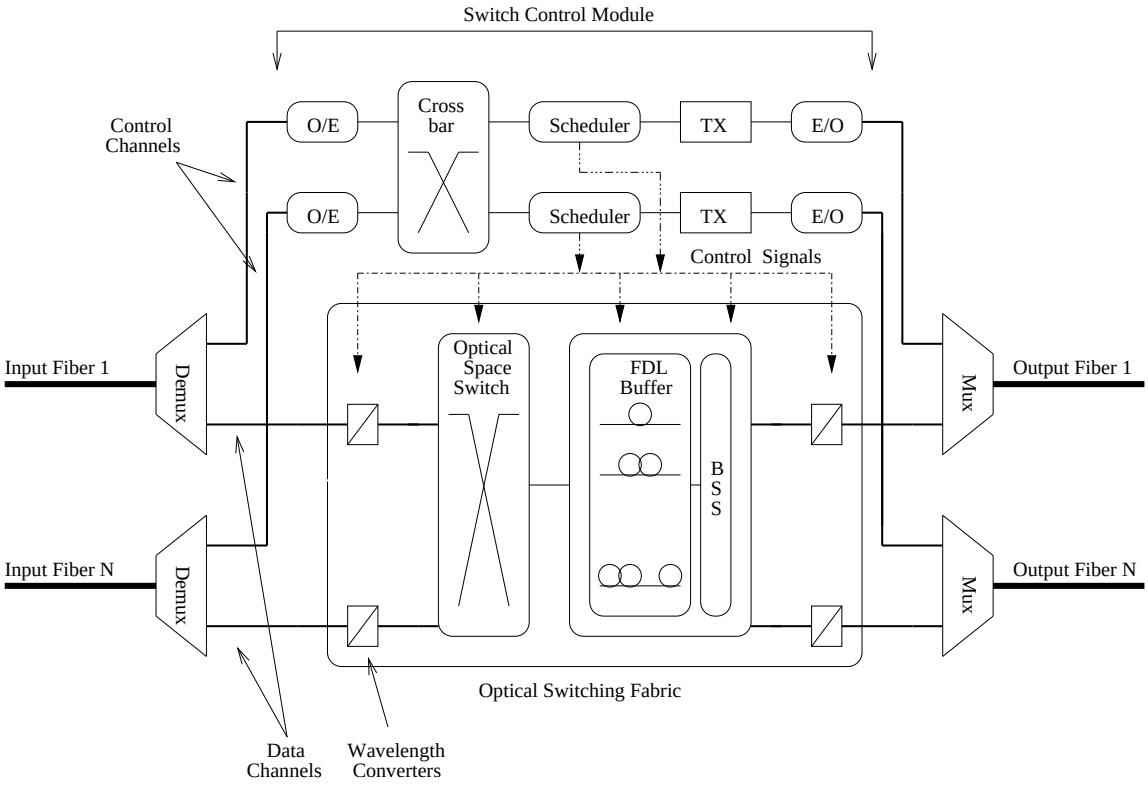


Figure 4.2. Block diagram of an OBS core node.

filling. Section 4.4 discusses the proposed segmentation-based scheduling algorithms with FDLs. Section 4.5 provides numerical results for the different scheduling algorithms. Section 4.6 concludes the chapter.

4.2 OBS Core Node Architecture

Figure 4.2 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

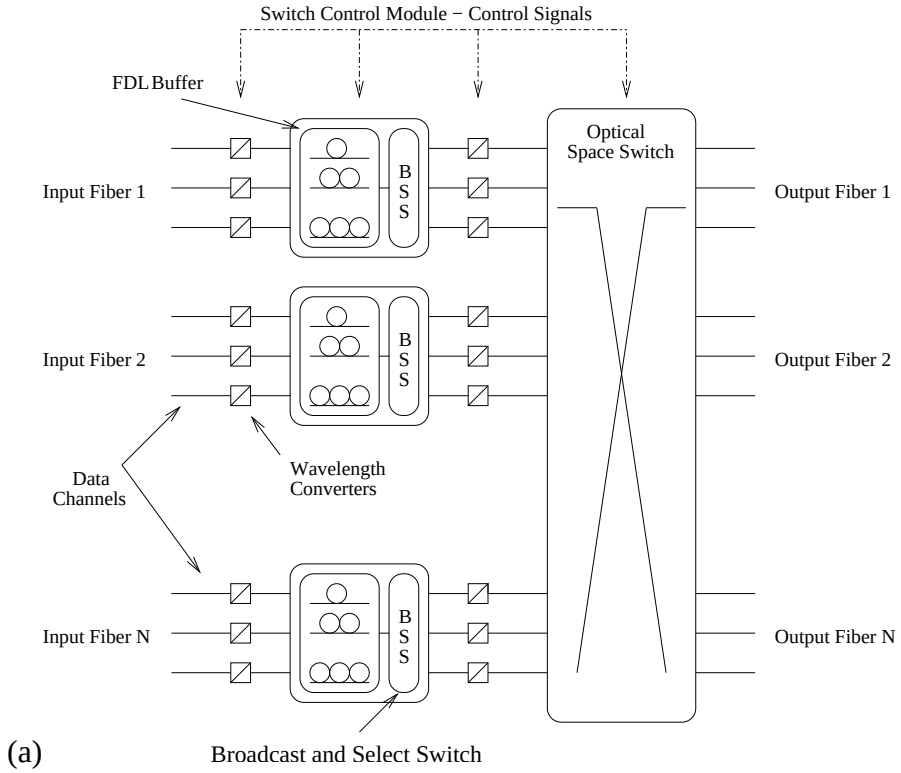
that the arrangement of the key components depends on the architecture of OBS node considered. A number of different OBS node architectures are possible using FDLs as optical buffers.

We consider two OBS node architectures with FDLs for realizing the proposed scheduling algorithms. The architecture in Fig. 4.3(a) shows an input-buffered OBS node with FDLs dedicated to each input port, while Fig. 4.3(b) shows an output-buffered OBS node with FDLs dedicated to each output port.

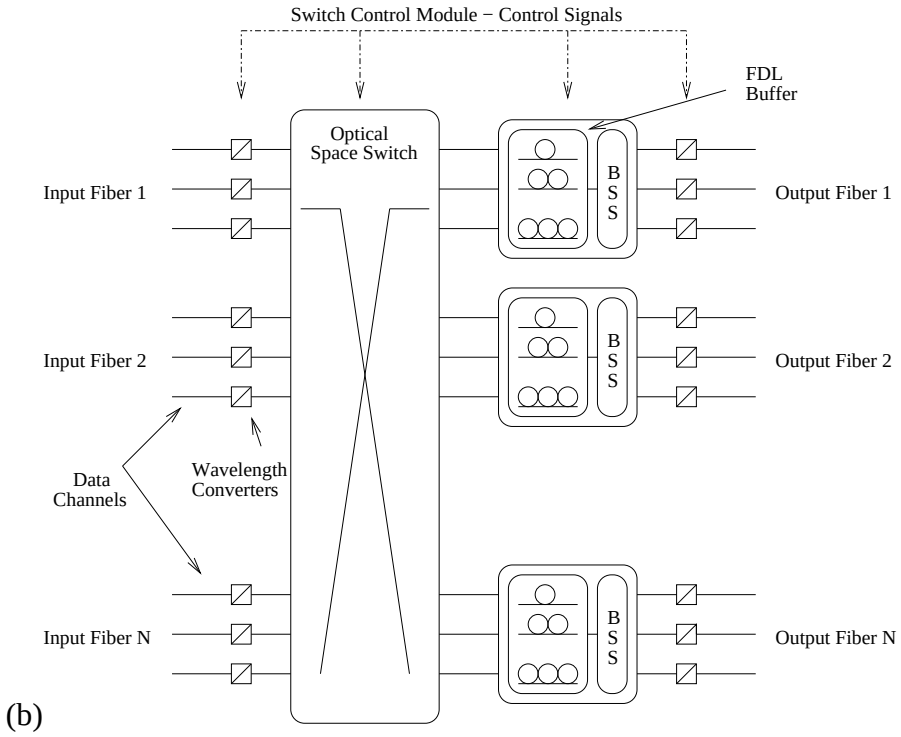
In the input-buffered OBS node architecture shown in Fig. 4.3(a), each input port is equipped with an FDL buffer containing N delay lines. The input-buffered architecture supports the *delay-first* scheduling algorithms. The n data channels are demultiplexed from each input fiber link and are passed 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. 4.3(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.



(a)



(b)

Figure 4.3. (a) Input-Buffer FDL Architecture, and (b) Output-Buffer FDL Architecture.

4.3 Segmentation-Based Non-Preemptive Scheduling Algorithms

The algorithm may need to maintain the 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:

- 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.

4.3.1 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 scheduling algorithm considers all outgoing data channels and calculates the overlap on every channel and chooses the data channel with minimum overlap.

For example, applying the NP-MOC algorithm to the example in Fig. 4.4(a), we see that data channel D_2 has the minimum loss, and the unscheduled burst is scheduled on D_2 (Fig. 4.5(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 $O(\log W)$.

4.3.2 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 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

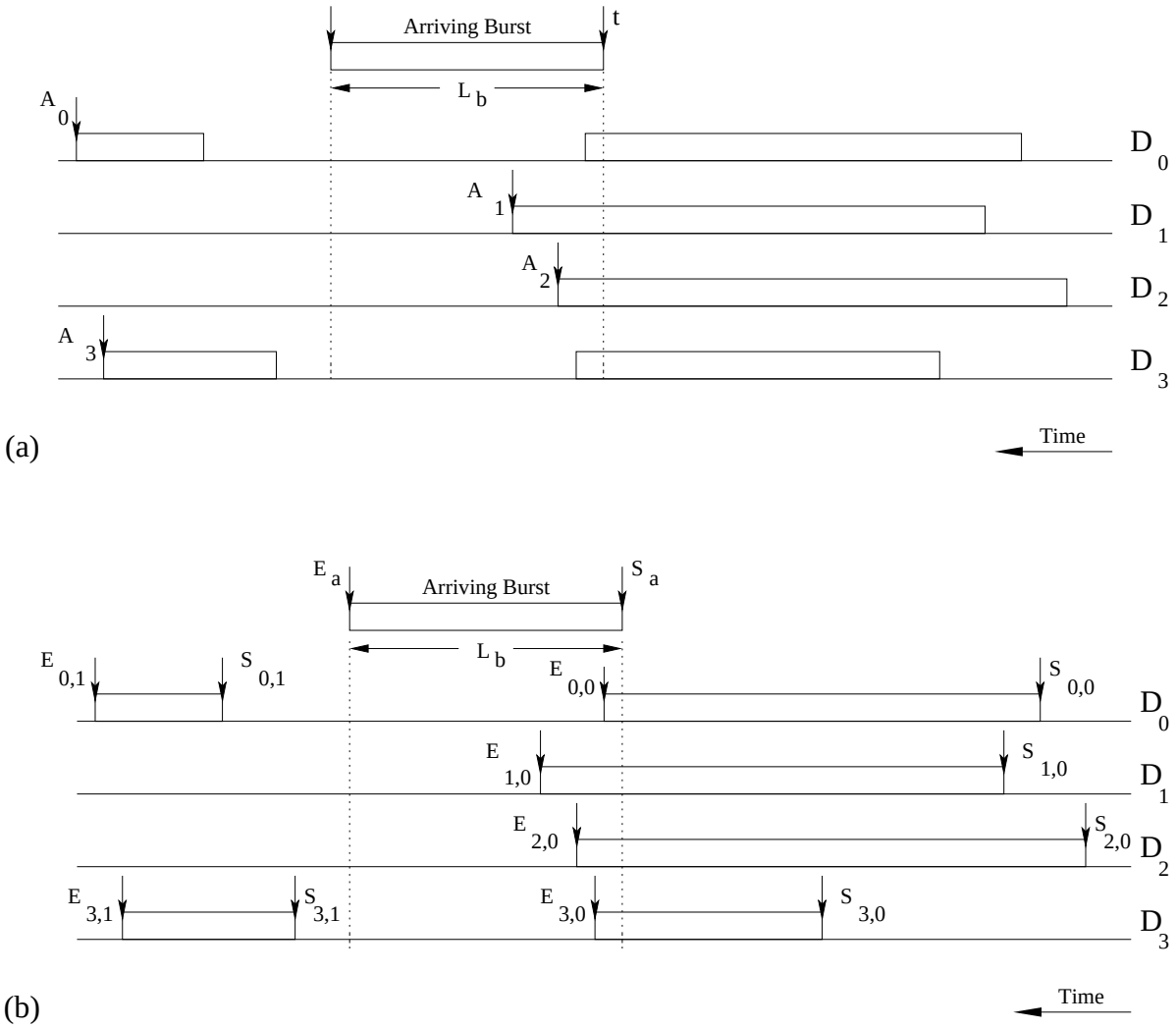


Figure 4.4. Initial data channel assignment using a) non-void filling and b) void filling scheduling.

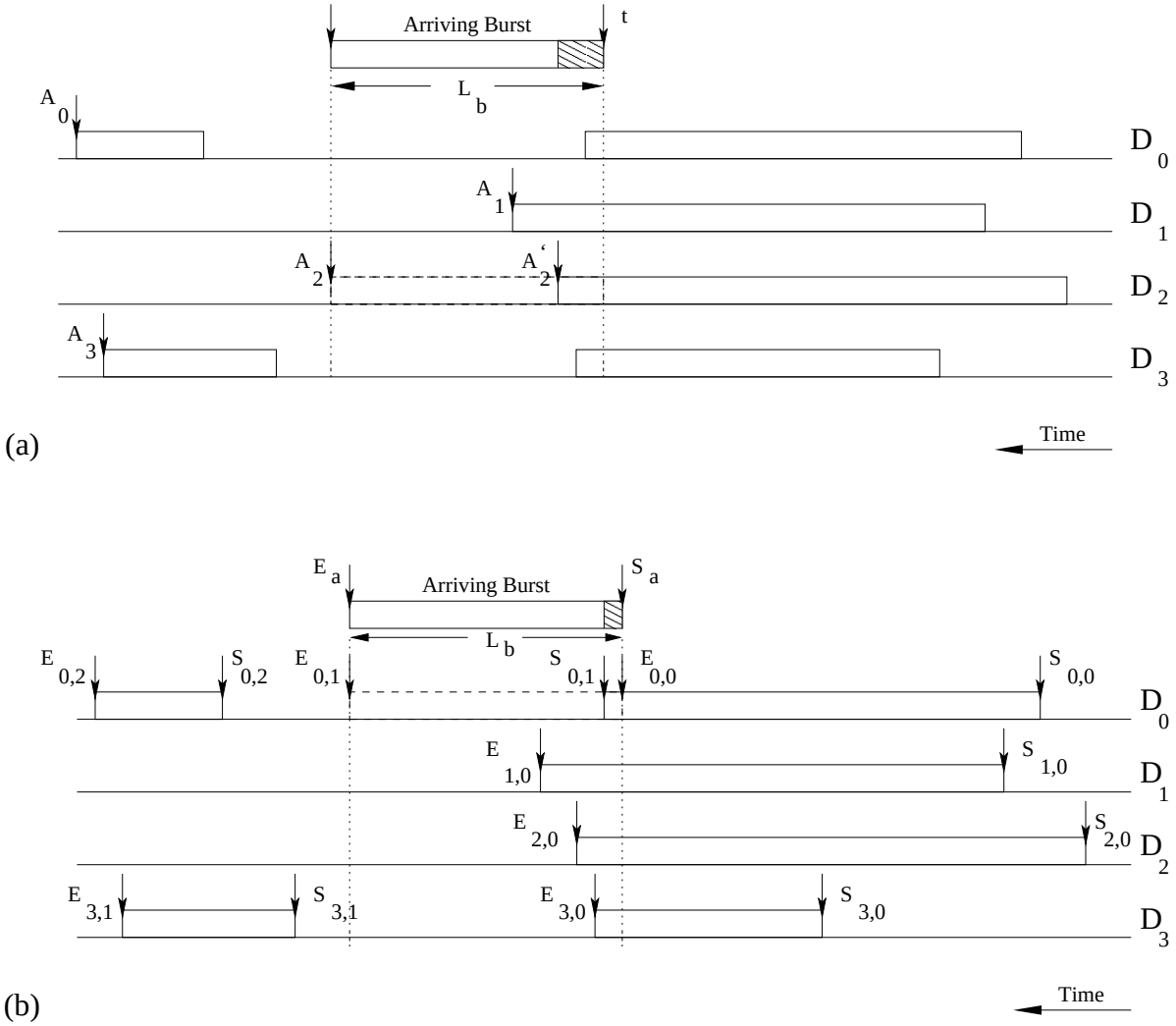


Figure 4.5. Illustration of non-preemptive (a) NP-MOC scheduling algorithm, and (b) NP-MOC-VF scheduling algorithm.

Table 4.1. Comparison of Segmentation-based Non-preemptive Scheduling Algorithms

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

is assigned to the unscheduled burst. For example, applying the NP-MOC-VF algorithm to the example in Fig. 4.4(b), we see that data channel D_0 has the minimum overlap, and the unscheduled burst is scheduled on D_0 (Fig. 4.5(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 $O(\log(W N_b))$.

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

4.4 Segmentation-Based Non-Preemptive Scheduling Algorithms with FDLs

There has been substantial work on scheduling using FDLs in OBS [28, 21, 78]. In this section, we propose a number of segmentation-based non-preemptive scheduling algorithms incorporating FDLs. Based on the two FDL architectures presented in Section 4.2, 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. In general, segment-first schemes will incur lower delays than delay-first schemes. In both the schemes, the scheduler 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.

4.4.1 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 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. 4.6(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.

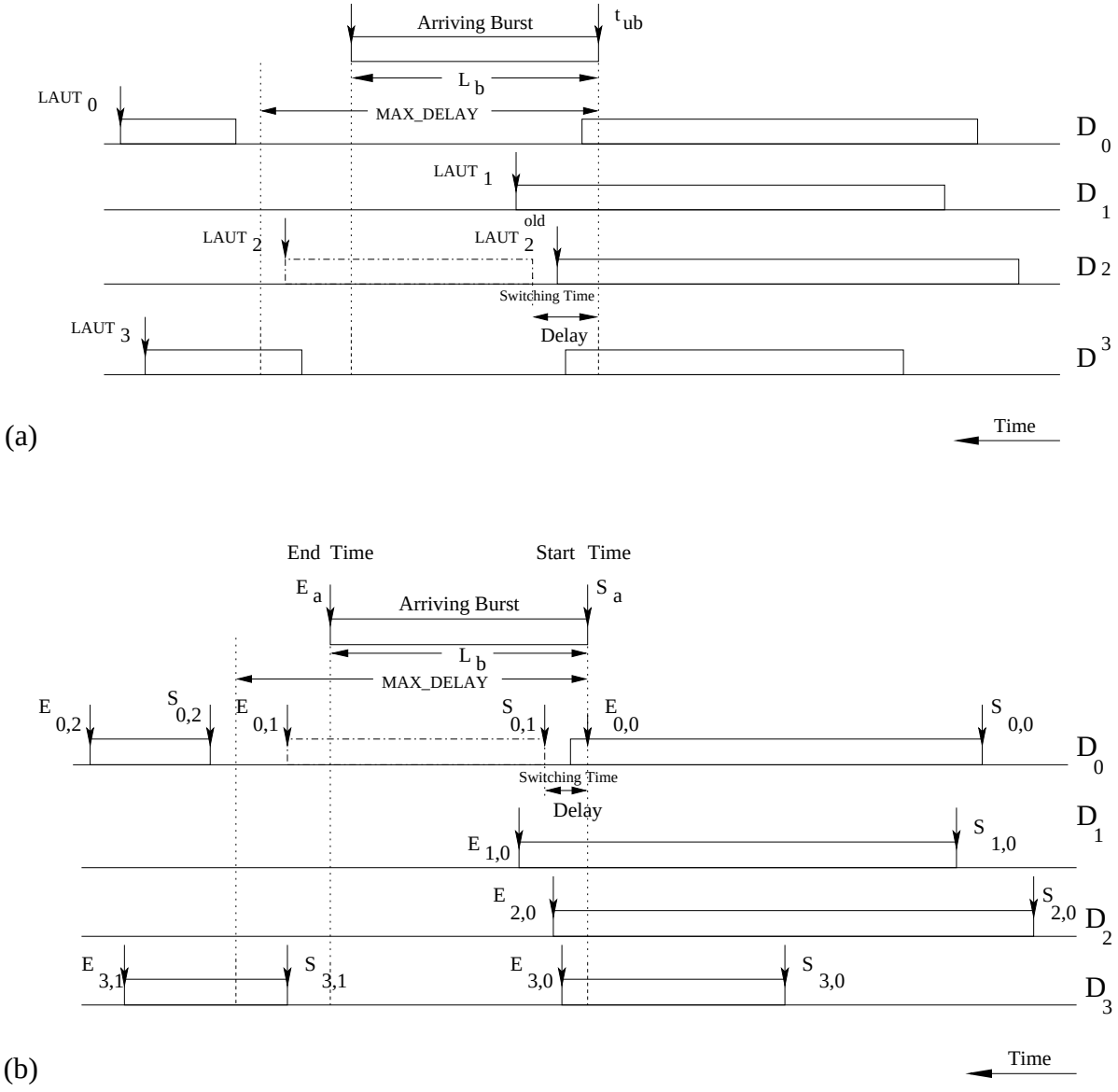


Figure 4.6. Illustration of (a) NP-DFMOC algorithm, and (b) NP-DFMOC-VF algorithm.

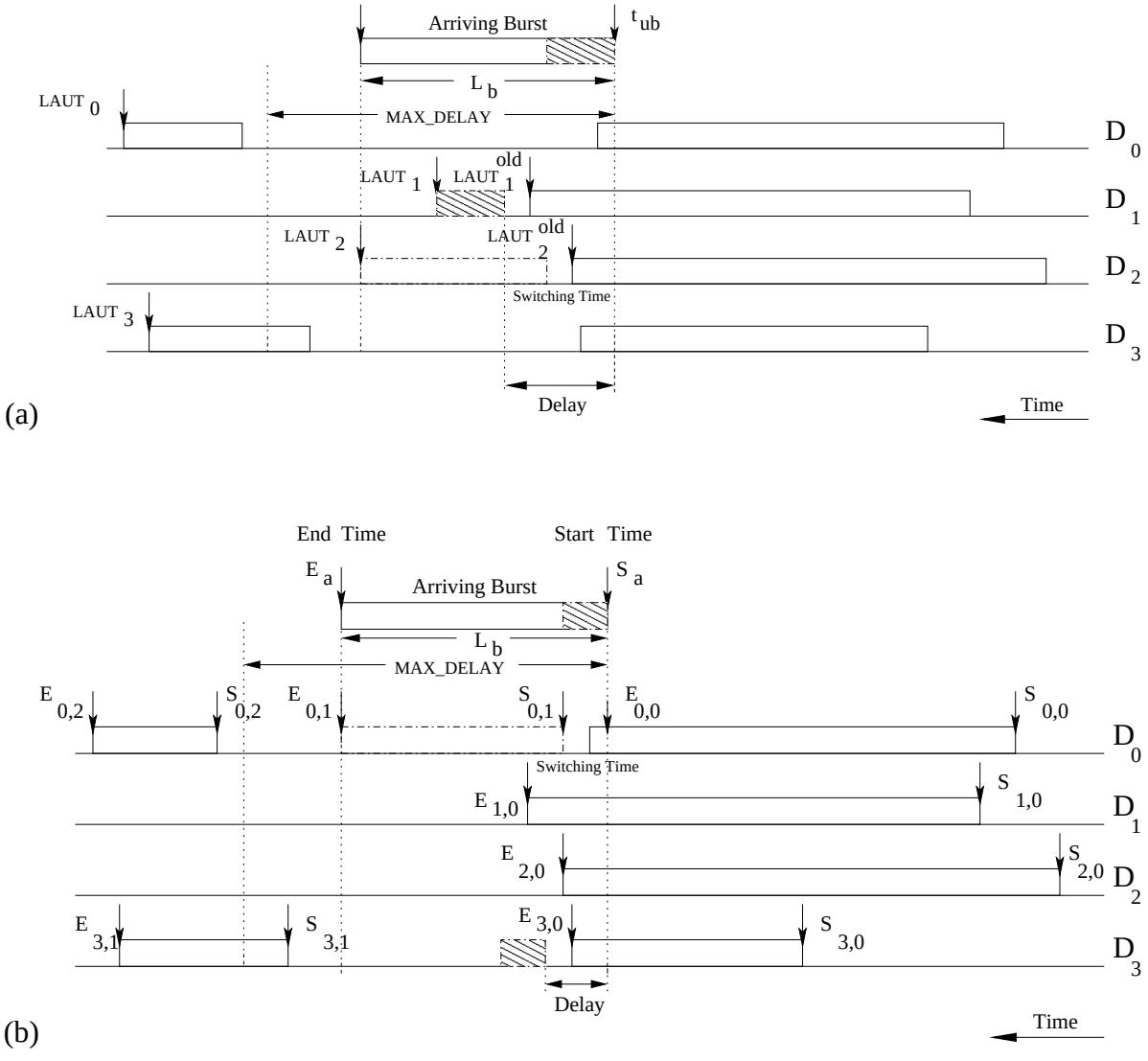


Figure 4.7. Illustration of (a) NP-SFMOC algorithm, and (b) NP-SFMOC-VF algorithm.

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. 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. 4.6(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.

4.4.2 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. 4.7(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. 4.7(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 4.2. 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

Table II 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.

In order to implement prioritized scheduling, we need to consider the overlap information based on the priority of the burst. In general, for burst of Priority i, the scheduler has to maintain a $Preemptive\_Olap_{ik}$ and a $Non\_Preemptive\_Olap_{ik}$ for every data channel k. For every data channel k, $Preemptive\_Olap_{ik}$ is given by the $Olap_{ik}$ of all bursts of priority j, where $j < i$ and $Non\_Preemptive\_Olap_{ik}$ is given by the $Olap_{ik}$ of all bursts of priority j, where $j \geq i$.

Without loss of generality, let us consider a two-priority network with Priority 0 bursts being higher priority then Priority 1 bursts. Table III gives the scheduling options that the scheduler must consider before selecting a wavelength for the arriving burst. We can see that on a Priority 0 burst arrival, for every wavelength k, the scheduler must compute $Non\_Preemptive\_Olap_{ik}$ from all the overlapping Priority 0 bursts and $Preemptive\_Olap_{ik}$

Table 4.3. Scheduling Options

Scheduled Arriving	Priority 0 Burst	Priority 1 Burst
Priority 0	$Non\_Preemptive\_Olap_{ik}$	$Preemptive\_Olap_{ik}$
Priority 1	$Non\_Preemptive\_Olap_{ik}$	$Non\_Preemptive\_Olap_{ik}$

from all the overlapping Priority 1 bursts.

4.5 Numerical Results

In order to evaluate the performance of the proposed channel scheduling algorithms, a simulation model is developed. Burst arrivals to the network are Poisson, and each 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) [131] to a few ns (SOA-based) [147]. 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. 4.8, where link distances are in km.

Figure 4.9(a) plots the total packet loss probability versus load for different channel

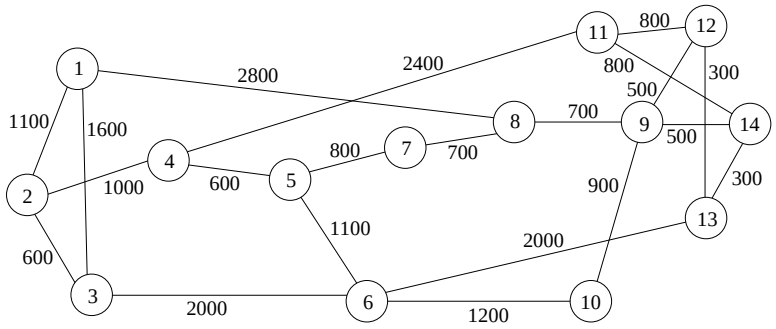
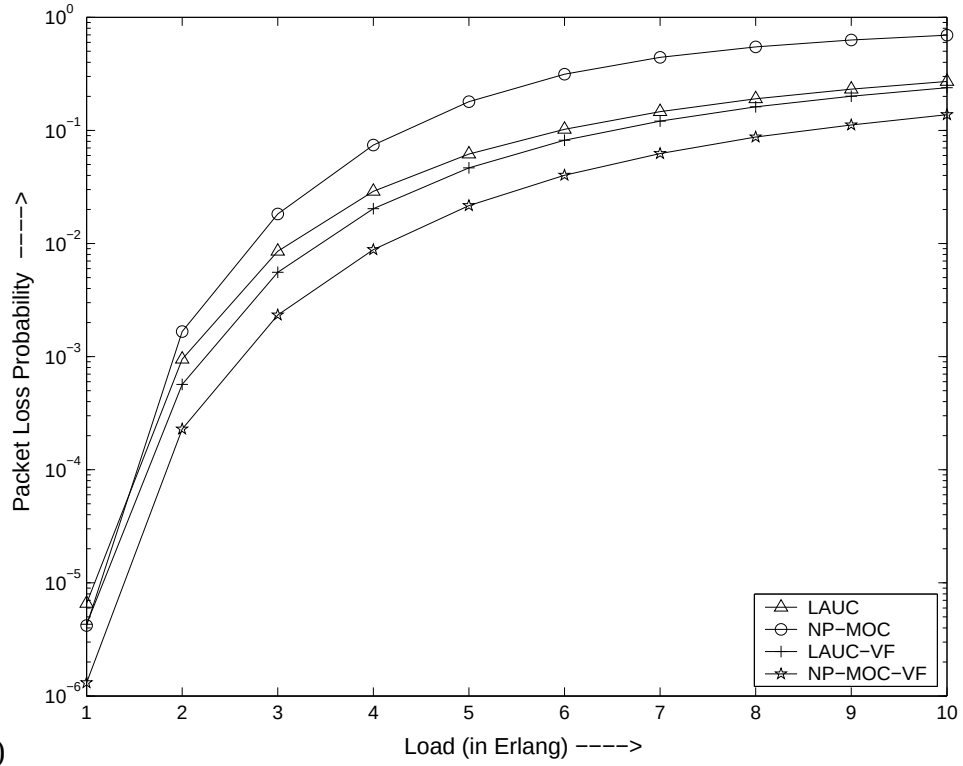


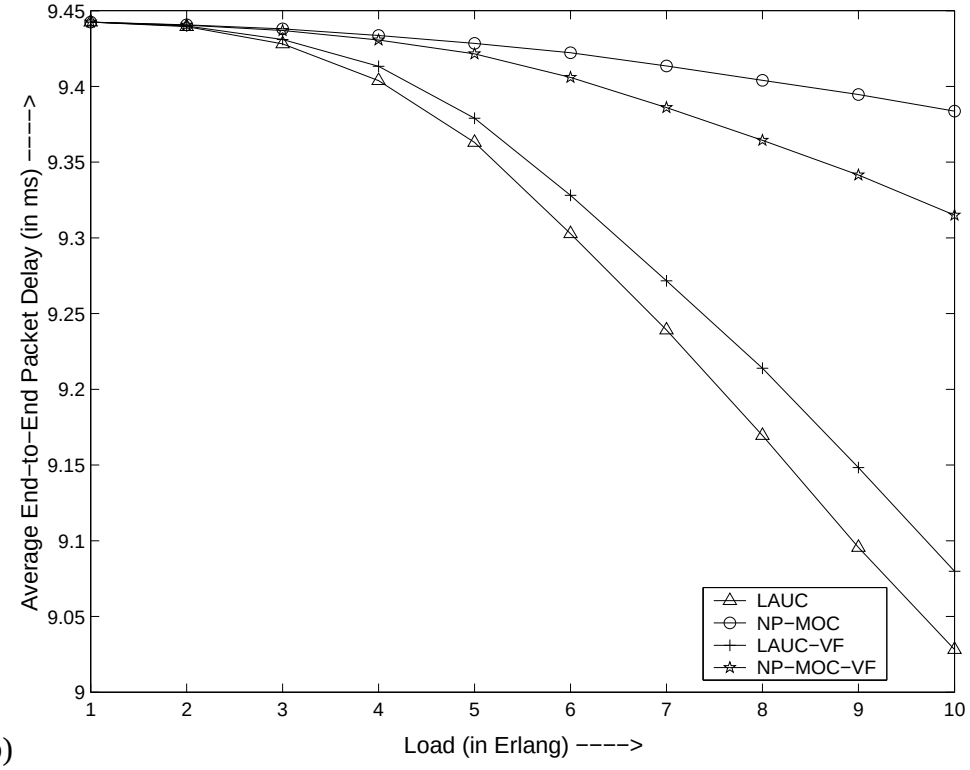
Figure 4.8. 14-Node NSF Network.

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 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 4.9(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



(a)



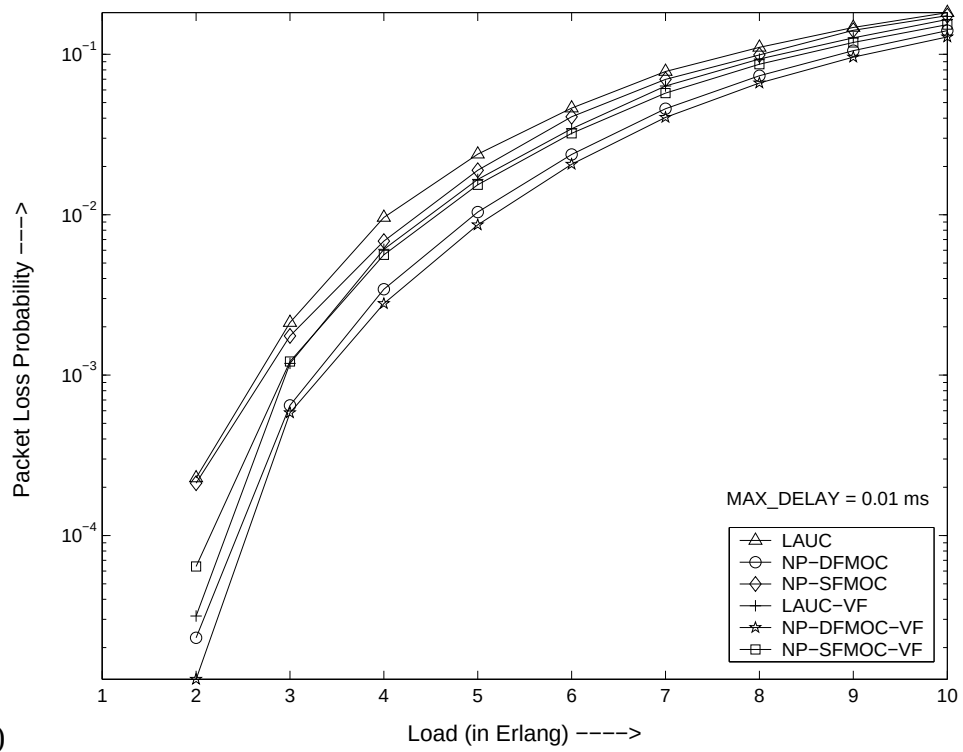
(b)

Figure 4.9. (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.

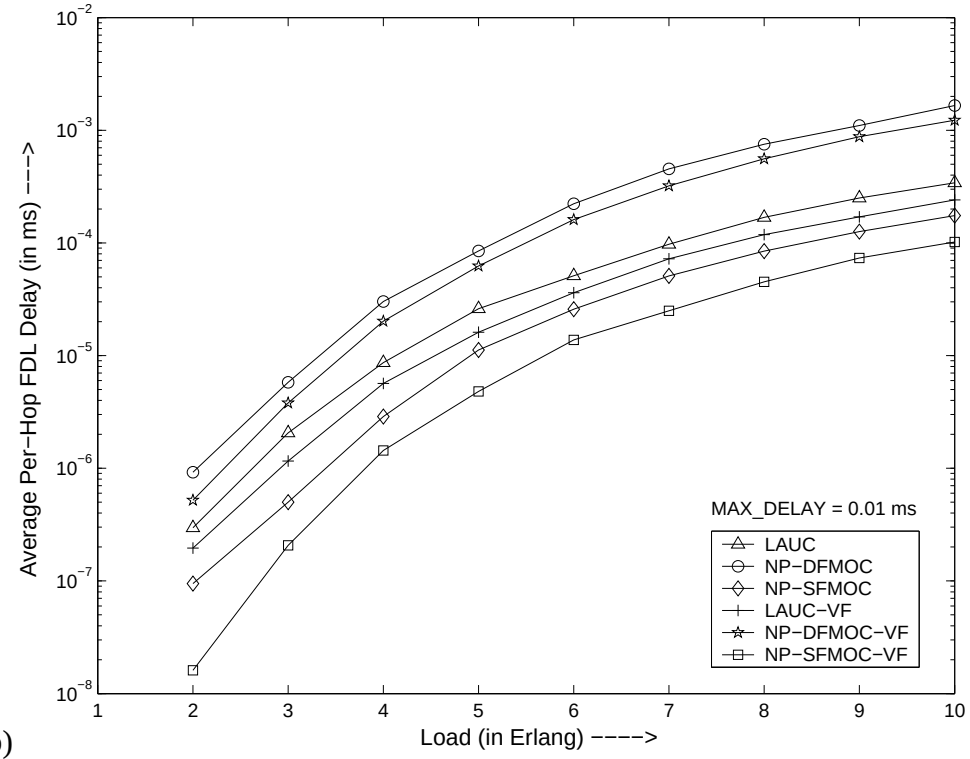
results in lower average end-to-end packet delay. We see that the NP-MOC algorithm has higher delay than the LAUC algorithm. Also, the NP-MOC-VF algorithm has higher delay than the LAUC-VF algorithm. We can also observe that LAUC has the least average end-to-end packet delay among all the algorithms.

Figure 4.10(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 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 4.10(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



(a)



(b)

Figure 4.10. (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.

compared to scheduling the bursts at the end of the horizon (LAUT) in the case of non-void filling algorithms. 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 [80].

4.6 Conclusion

In this chapter, we considered burst segmentation and FDLs with wavelength conversion for burst scheduling in optical burst-switched networks, and we proposed 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, while the 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.

CHAPTER 5

PRIORITIZED BURST SEGMENTATION FOR PROVIDING QOS IN OPTICAL BURST-SWITCHED NETWORKS

5.1 Introduction

An important issue in optical burst-switched networks is how to provide differentiated service in order to support the various quality of service (QoS) requirements of different applications.

In this chapter, we focus on the issue of providing QoS support in OBS through prioritized contention resolution. Prioritized contention resolution is provided using prioritized burst segmentation and prioritized deflection routing. In order to implement the prioritized contention resolution schemes, priority values have to be included as a field in the burst header packet (BHP). This priority field is used to preferentially segment and deflect bursts when resolving contentions in the core. We develop analytical and simulation models to evaluate the packet loss probability of the various QoS schemes. In this work, we assume that JET signaling is used and that there are no fiber delay lines or wavelength converters in the network. Without loss of generality, we assume that there are two priority classes supported in the OBS network, and that a high-priority burst is one which has low delay and loss tolerance while a low-priority burst has relaxed delay and loss constraints.

The remainder of the chapter is organized as follows. Section 5.2 discusses the prioritized contention resolution policies employing burst segmentation and deflection. In Section 5.3, we develop an analytical model to calculate the packet loss probability for the proposed prioritized burst segmentation. Section 5.4 provides numerical results from simulation and analysis and compares the results of the different prioritized contention resolution policies. Section 5.5 concludes the chapter.

5.2 Prioritized Contention Resolution

To overcome some of the limitations of OBS, burst segmentation can be used to minimize packet loss during contention. In burst segmentation, a burst is divided into multiple segments, and when contention occurs, only those segments of a given burst which overlap with segments of another burst will be dropped. If switching time is non-negligible, then additional segments may be lost when the output port is switched from one burst to another. Segmentation can be used to minimize loss of packets during a contention, and can also allow high-priority bursts to preempt low-priority bursts. In these discussions, the burst which arrives at a node first will be referred to as the *original* burst, and the burst which arrives later will be referred to as the *contending* burst. There are two approaches for segmenting a burst when contention occurs. The first approach is to segment the tail of the original burst, and the second approach is to segment the head of the contending burst. A significant advantage of segmenting the tail of bursts rather than segmenting the head 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. In this chapter, we will assume that the remaining tail of the original burst will be dropped when segmentation takes place. Also, when a burst is segmented, its control message is updated accordingly.

Burst segmentation can also be implemented with deflection. Rather than dropping the tail segment of the original burst, we can either deflect the entire contending burst, or we can deflect the tail segment of the original burst. Implementing segmentation with deflection increases the probability that a burst's packets will reach the destination, and hence improves performance. At each node, one or more alternate deflection ports can be specified for each destination. The order in which the alternate deflection ports are attempted is determined by a shortest-path policy.

The foundation for providing QoS in IP over OBS networks is service differentiation in the OBS core. We introduce and evaluate a new approach for such differentiation based on the concepts of burst segmentation and burst deflection. Burst segmentation enables the

contending burst to preempt the original burst; hence, we have a choice of dropping either the contending burst or segmenting the original burst during a contention. Bursts are assigned priorities which are stored in the BHP, and contention between bursts is resolved through selective segmentation, deflection, and burst dropping based on these priorities.

We approach the general problem by first defining the possible segmentation and deflection policies which can be applied when a contention occurs. We then define the possible contention scenarios which can take place between bursts of different priorities and lengths. Finally, we specify which policy to apply for each specific contention scenarios.

When two bursts contend with one another, one of five policies described in Section 3.3, namely DP, SDP, DDP, SDDP, and DSDP may be applied to resolve the contention:

We consider a total of four different 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, we can break the tie 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, we specify one of the contention resolution policies described above.

Figure 5.1 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, DDP 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, SDDP 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 DDP policy is applied. If the contending burst is longer

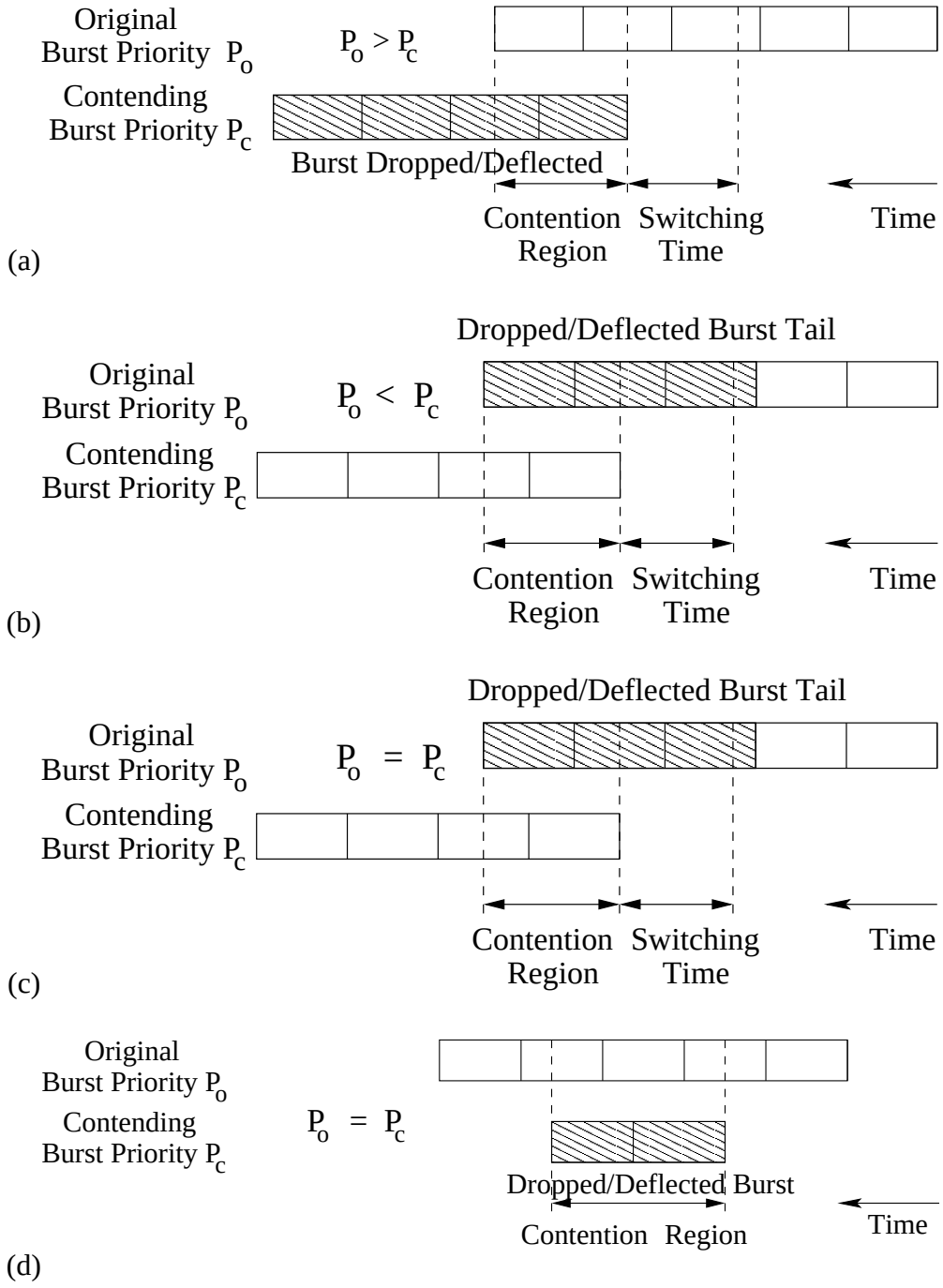


Figure 5.1. (a) Contention of a low-priority burst with a high-priority burst. (b) Contention of a high-priority burst with a low-priority burst. (c) Contention of two equal-priority bursts with longer contending burst. (d) Contention of two equal-priority bursts with shorter contending burst.

Table 5.1. QoS schemes.

Priority	Length	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5
$P_o > P_c$	any	DDP	DDP	DP	DDP	DP
$P_o < P_c$	any	SDDP	SDDP	SDP	DDP	DP
$P_o = P_c$	$L_o > L_c$	DDP	DDP	DP	DDP	DP
$P_o = P_c$	$L_o < L_c$	DSDP	SDDP	SDP	DDP	DP

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 entire contending burst; thus, either DSDP or SDDP may be applied. We consider both options, referring to the scheme in which DSDP is applied as Scheme 1, and the scheme in which SDDP is applied as Scheme 2. For comparison, we further define schemes which do not take advantage of either segmentation or deflection. In Scheme 3, segmentation is supported but deflection is not, while in Scheme 4, deflection is supported but segmentation is not. In Scheme 5, neither deflection nor segmentation are supported. These schemes are 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 and L_c refer to the remaining length of the original burst and the length of the contending burst respectively.

5.3 Analytical Model

In this section, we develop an analytical model for evaluating the packet loss probabilities with prioritized burst segmentation. We evaluate a modified version of Scheme 3 in which no burst length comparison is done. If two bursts are of equal-priority, we give priority to the contending burst. We assume that high and low-priority bursts arrive to the network according to a Poisson process with rate λ^{sd} and γ^{sd} bursts per second for source-destination pair sd respectively. Fixed routing is assumed, and no buffering is supported at core nodes. We also assume that all bursts have the same offset time. This implies that the BHP of the original burst always arrives before the BHP of the contending burst. Traffic on each link is assumed

to be independent. Without loss of generality, we consider a two-priority OBS network such that, Priority 0 bursts have higher priority that Priority 1 bursts. First, we analyze the packet loss probability for the high-priority bursts. We begin by defining the following notation:

- λ_l^{sd}: arrival rate of high-priority bursts to link l, on the path between source s and destination d.

- γ_l^{sd}: arrival rate of low-priority bursts to link l, on the path between source s and destination d.

- $\lambda_l = \sum_{sd} \lambda_l^{sd}$: arrival rate of high-priority bursts to link l, due to all source-destination pairs sd.

- $\gamma_l = \sum_{sd} \gamma_l^{sd}$: arrival rate of low-priority bursts to link l, due to all source-destination pairs sd.

- r_{sd}: route from source s to destination d.

The load placed on a link l by traffic going from source s to destination d depends on whether link l is on the path to destination d. If link l is on the path to d, then the load applied to link l by sd traffic is simply λ^{sd}. Thus,

$$\lambda_l^{sd} = \begin{cases} \lambda^{sd} & \text{if } l \in r_{sd} \\ 0 & \text{if } l \notin r_{sd}. \end{cases} \tag{5.1}$$

Also, the total high-priority (new) burst arrival into the network, λ, is given by:

$$\lambda = \sum_s \sum_d \lambda^{sd}. \tag{5.2}$$

We calculate the packet loss probability by finding the distribution of the burst length at the destination and comparing the mean burst length at the destination to the mean burst length at the source. Let the initial cumulative distribution function of the burst length be $G^0_{l_0^{sd}}(t)$ for high-priority bursts transmitted from source s to destination d, where l_0^{sd} is the zeroth hop link between source s to destination d. The cumulative distribution function of the burst after

k hops is $G^0_{l^{sd}_k}(t)$. Let $F^0_{l^{sd}_k}(t)$ be the cumulative distribution function for the arrival time of the next high-priority burst on the k^{th} hop link l between source-destination pair sd:

$$F^0_{l^{sd}_k}(t) = \quad 1 - e^{-\lambda_{l^{sd}_k} t}, \tag{5.3}$$

where $\lambda_{l^{sd}_k}$ is the arrival rate of all high-priority bursts on the k^{th} hop link of the path between source s and destination d.

We note that a high-priority burst is segmented only if the next arriving burst is also of high-priority, but is not affected by the arrival of low-priority bursts. The burst length will be reduced if another high-priority burst arrives while the original burst is being transmitted; thus, the probability that the burst length is less than or equal to t after the first hop is equal to the probability that the initial burst length is less than or equal to t or that the next high-priority burst arrives in time less than or equal to t. Therefore,

$$\begin{aligned} G^0_{l^{sd}_1}(t) &= \quad 1 - (1 - G^0_{l^{sd}_0}(t))(1 - F^0_{l^{sd}_1}(t)) \\ &= \quad 1 - (1 - G^0_{l^{sd}_0}(t)) e^{-\lambda_{l^{sd}_1} t}. \end{aligned} \tag{5.4}$$

Similarly, let $G_2(t)$ be the cumulative distribution function of the burst after the second hop:

$$\begin{aligned} G^0_{l^{sd}_2}(t) &= \quad 1 - (1 - G^0_{l^{sd}_1}(t))(1 - F^0_{l^{sd}_2}(t)) \\ &= \quad 1 - (1 - G^0_{l^{sd}_0}(t)) e^{-(\lambda_{l^{sd}_1} + \lambda_{l^{sd}_2}) t}. \end{aligned} \tag{5.5}$$

In general,

$$\begin{aligned} G^0_{l^{sd}_k}(t) &= \quad 1 - (1 - G^0_{l^{sd}_{k-1}}(t))(e^{-\lambda_{l^{sd}_k}}) \\ &= \quad 1 - (1 - G^0_{l^{sd}_0}(t)) e^{-\left(\sum_{i=1}^{k} \lambda_{l^{sd}_i}\right) t}. \end{aligned} \tag{5.6}$$

We now find the expected length after k hops and compare this length with the expected length at the source node in order to obtain the expected loss that a particular burst will experience. Let $L^0_{l^{sd}_k}$ be the expected length of the high-priority burst at the k^{th} hop.

Case (1): If we have fixed-sized bursts of length, $\frac{1}{\mu} = T^0$, then the initial distribution of the burst length is given by:

$$G^0_{l^{sd}_0}(t) = Pr(T \leq t) = \begin{cases} 1 \text{ if } t \geq T^0 \\ 0 \text{ if } t < T^0. \end{cases} \tag{5.7}$$

Substituting (5.7) into (5.6) and taking the expected value, we obtain:

$$L^0_{l^{sd}_k} = \frac{1 - e^{-\sum_{i=1}^{k} \lambda_{l^{sd}_i} T^0}}{\sum_{i=1}^{k} \lambda_{l^{sd}_i}}. \tag{5.8}$$

Case (2): If the initial burst length is exponentially distributed, we have:

$$G^0_{l^{sd}_0}(t) = 1 - e^{-\mu t}. \tag{5.9}$$

Substituting (5.9) into (5.6) and taking the expected value, we obtain:

$$L^0_{l^{sd}_k} = \frac{1}{\sum_{i=1}^{k} \lambda_{l^{sd}_i} + \mu}. \tag{5.10}$$

We now find the expected length after K hops, where K is the total number of hops between s and d. Let $Loss^0_{sd}$ be the expected length of the burst lost per high-priority burst for a burst traveling from s to d:

$$Loss^0_{sd} = \frac{1}{\mu} - L^0_{l^{sd}_K}. \tag{5.11}$$

Hence, the packet loss is proportional to the length of the route and the length of the burst. The packet loss probability of high-priority bursts, P^{sd}_{loss0}, is then given by:

$$\begin{aligned} P^{sd}_{loss0} &= \frac{E[Length\ Lost]}{E[Initial\ Length]} \\ &= Loss^0_{sd} \cdot \mu. \end{aligned} \tag{5.12}$$

We can then find the average packet loss probability of high-priority bursts for the system by finding the individual loss probability for each source-destination pair, and taking the weighted average of the loss probabilities:

$$P^0_{loss} = \sum_s \sum_d \frac{\lambda^{sd}}{\lambda} P^{sd}_{loss}. \tag{5.13}$$

We also calculate the average service time on a link l, where l is the k^{th} link from source s to destination d:

$$\frac{1}{\bar{\mu}_l} = \sum_{s,d:\ \lambda_l^{sd}>0} \frac{\lambda_l^{sd}}{\lambda_l} \cdot \frac{1}{\mu_{l_k^{sd}}}, \tag{5.14}$$

where, $\mu_{l_k^{sd}} = \frac{1}{L^0_{l_k^{sd}}}$.

Using $\bar{\mu}_l$, we can calculate the utilization for high-priority bursts on link l:

$$\rho_l = \frac{\lambda_l}{\bar{\mu}_l}. \tag{5.15}$$

Now, we calculate the probability of low-priority packet loss. The entire low-priority burst is dropped if a high-priority burst is occupying the channel. Thus, the arrival rate of low-priority bursts depends upon the link utilization of high-priority bursts. The offered load on the first hop is the total offered load from source to destination. On subsequent hops, the offered load is the load from the previous hop that was not blocked by high-priority traffic, thus,

$$\gamma_l^{sd} = \begin{cases} \gamma^{sd} & \text{if } l \in r_{sd},\ l = l_0^{sd} \\ \gamma_h^{sd}(1-\rho_h) & \text{if } l, h \in r_{sd},\ l = l_i^{sd},\ h = l_{i-1}^{sd},\ i \geq 1 \\ 0 & \text{if } l \notin r_{sd}. \end{cases} \tag{5.16}$$

The calculation of low-priority packet loss probability is similar to that of high-priority packet loss. Let the initial cumulative distribution function of the burst length be $G^1_{l_0^{sd}}(t)$, and the cumulative distribution function of the burst after k hops be $G^1_{l_k^{sd}}(t)$ for low-priority bursts transmitted from source s to destination d. Let $F^1_{l_k^{sd}}(t)$ be the cumulative distribution function for the arrival time of the next burst on the k^{th} hop link. Here we consider the total arrival rate of bursts of both high and low priorities:

$$F^1_{l_k^{sd}}(t) = \ 1 - e^{-(\lambda_{l_k^{sd}} + \gamma_{l_k^{sd}})t}, \tag{5.17}$$

where $\gamma_{l_k^{sd}}$ and $\lambda_{l_k^{sd}}$ are the arrival rates of all low and high-priority bursts on the k^{th} hop link of the path between source s and destination d.

The burst length will be reduced if another burst of any priority arrives while the original burst is being transmitted; thus, the cumulative distribution function after the first

hop is equal to the probability that the initial burst length is less than or equal to t or the next burst arrives in time less than or equal to t:

$$\begin{aligned} G^1_{l^{sd}_1}(t) = & \quad \rho_1 + (1-\rho_1)[1-(1-G^1_{l^{sd}_0}(t))(1-F^1_{l^{sd}_1}(t))] \\ = & \quad 1-(1-\rho_1)(1-G^1_{l^{sd}_0}(t))e^{-(\lambda_{l^{sd}_1}+\gamma_{l^{sd}_1})t}. \end{aligned} \tag{5.18}$$

Similarly, $G^1_{l^{sd}_2}(t)$ is the cumulative distribution function of the burst length after the second hop:

$$\begin{aligned} G^1_{l^{sd}_2}(t) = & \quad \rho_2 + (1-\rho_2)[1-(1-G^1_{l^{sd}_1}(t))(1-F^1_{l^{sd}_2}(t))] \\ = & \quad 1-(1-\rho_2)(1-\rho_1)(1-G^1_{l^{sd}_0}(t))e^{-(\lambda_{l^{sd}_2}+\lambda_{l^{sd}_1}+\gamma_{l^{sd}_2}+\gamma_{l^{sd}_1})t}. \end{aligned} \tag{5.19}$$

In general,

$$\begin{aligned} G^1_{l^{sd}_k}(t) = & \quad \rho_k + (1-\rho_k)[1-(1-G^1_{l^{sd}_{k-1}}(t))(1-F^1_{l^{sd}_k}(t))] \\ = & \quad 1-\Pi^k_{i=1}(1-\rho_i)(1-G^1_{l^{sd}_0}(t))e^{-\left(\sum^k_{j=1}\lambda_{l^{sd}_j}+\gamma_{l^{sd}_j}\right)t}. \end{aligned} \tag{5.20}$$

We now find the expected length after k hops and compare with the expected length at the source node to obtain the expected loss. Let $L^1_{l^{sd}_k}$ be the expected length of the low-priority burst at the k^{th} hop.

Case (1): If we have fixed-sized bursts of length, $\frac{1}{\mu} = T^1$, the initial distribution of the burst length is given by:

$$G^1_{l^{sd}_0}(t) = Pr(T \leq t) = \begin{cases} 1 \text{ if } t \geq T^1 \\ 0 \text{ if } t < T^1. \end{cases} \tag{5.21}$$

Therefore $L^1_{l^{sd}_k}$ is given by:

$$L^1_{l^{sd}_k} = \frac{\Pi^k_{i=1}(1-\rho_i)\left(1-e^{-\sum^k_{i=1}\left(\lambda_{l^{sd}_i}+\gamma_{l^{sd}_i}\right)T^1}\right)}{\sum^k_{i=1}\left(\lambda_{l^{sd}_i}+\gamma_{l^{sd}_i}\right)}. \tag{5.22}$$

Case (2): If the initial burst length is exponentially distributed, we have:

$$G^1_{l^{sd}_0}(t) = 1-e^{-\mu t}. \tag{5.23}$$

Therefore $L^1_{l^{sd}_k}$ is given by:

$$L^1_{l^{sd}_k} = \frac{\Pi^k_{i=1}(1-\rho_i)}{\sum^k_{j=1}\left(\lambda_{l^{sd}_j}+\gamma_{l^{sd}_j}\right)+\mu}. \tag{5.24}$$

Let $Loss^1_{sd}$ be the expected length of the burst lost per low-priority burst for a burst traveling from s to d:

$$Loss^1_{sd} = \frac{1}{\mu} - L^1_{l^{sd}_K}. \tag{5.25}$$

The probability of packet loss for low-priority bursts is given by:

$$P^{sd}_{loss1} = Loss^1_{sd} \cdot \mu. \tag{5.26}$$

We can then find the average packet loss probability of low-priority bursts for the system by finding the individual loss probability for each source-destination pair, and taking the weighted average of the loss probabilities:

$$P^1_{loss} = \sum_s \sum_d \frac{\gamma^{sd}}{\gamma} P^{sd}_{loss1}. \tag{5.27}$$

Note that, if two contending bursts follow the same route, then the original burst will only be segmented at the first instance of contention; However, the model assumes that the arrivals of the two contending bursts are uncorrelated on the subsequent links in the route. Thus, the model over-estimates the packet loss.

Also, if a burst is segmented in the middle of a packet, the model does not account for the entire packet loss, which leads to a slight under-estimation of packet loss. However, this under-estimation of packet loss is insignificant compared to the over-estimation of the packet loss due to the uncorrelated arrival assumption.

This analysis may be extended to any arbitrary number of priorities in a straightforward manner. Also, a more accurate model may be obtained by using a reduced load approximation for the arrival of the low-priority bursts and by taking into account the link correlation effect [148, 149, 140].

5.4 Numerical Results

In order to evaluate the performance of the proposed schemes and to verify the analytical models, a simulation model is developed. Burst arrivals to the network are assumed to be Poisson with rate λ. Burst lengths are exponentially distributed with average length of $1/\mu$ = 100 ms. The link transmission rate is 10 Gb/s. Packets are assumed to be 1250 bytes and each segment consists of a single packet. The configuration time of the switching is assumed to be 10 μs. There is no buffering or wavelength conversion at the core nodes. Burst arrivals are uniformly distributed over all sender-receiver pairs, and shortest-path routing is assumed. Figure 5.2 shows the 14-node NSF network on which the simulation is implemented.

5.4.1 Analytical Results

Let us consider a network with two priorities. The fraction of high-priority (Priority 0) traffic is 20%, and the fraction of low-priority (Priority 1) traffic is 80%. In the analytical model, we ignore the switching time and header processing time.

Figure 5.3 plots the packet loss probability versus load for high-priority and low-priority packets for Scheme 1, with exponential burst length, and for fixed-sized bursts. In Scheme 1, the contending burst preempts the original burst if the contending burst is of equal or higher priority, otherwise, the contending burst is dropped. We observe that the analytical model slightly over-estimates the packet loss probabilities due to the independent link assumption. We also observe that the packet loss with fixed-sized bursts is lower than packet loss with exponentially distributed burst sizes, since the maximum number of packets lost per contention is potentially less with a fixed initial burst size. This observation may be useful when determining the burst assembly policy.

5.4.2 Simulation Results

Figure 5.4 plots the packet loss probability versus load for high-priority (Priority 0) and low-priority (Priority 1) packets for Scheme 1 through Scheme 5, with fixed-sized bursts. The

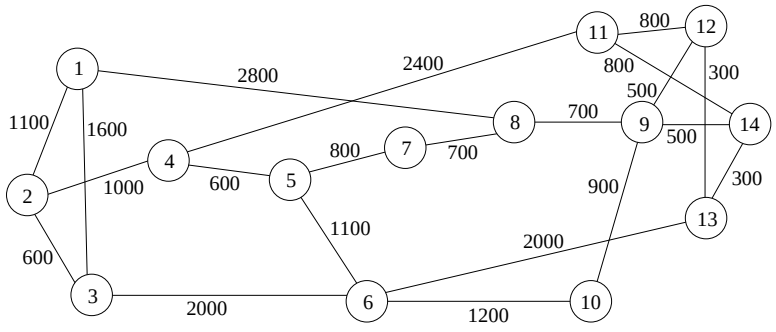


Figure 5.2. Picture of NSF network with 14 nodes (distance in km).

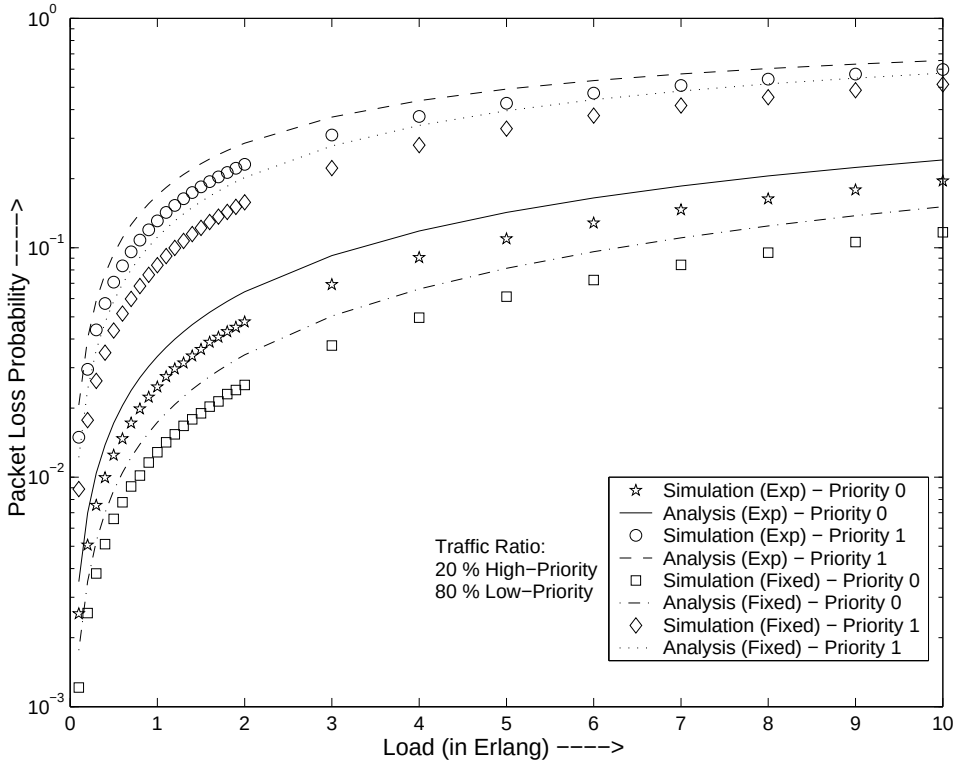


Figure 5.3. Packet loss probability versus load for both exponential initial burst size, $1/\mu$ = 100 ms and fixed initial burst size = 100 packets, using Scheme 3 without burst length comparison.

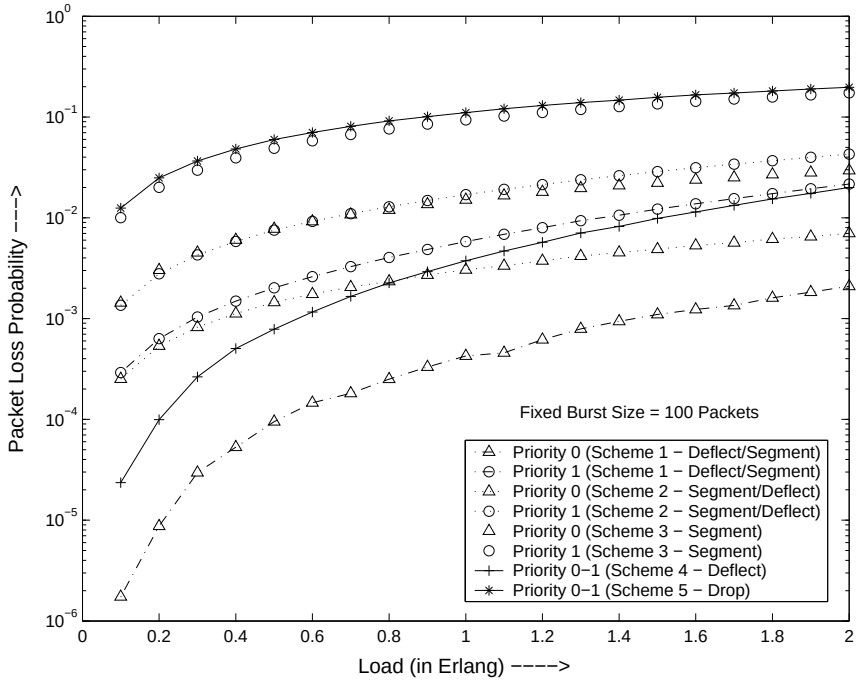


Figure 5.4. Packet loss probability versus load for different QoS schemes with fixed burst size = 100 packets, with the traffic ratio being 20% Priority 0 and 80% Priority 1 bursts.

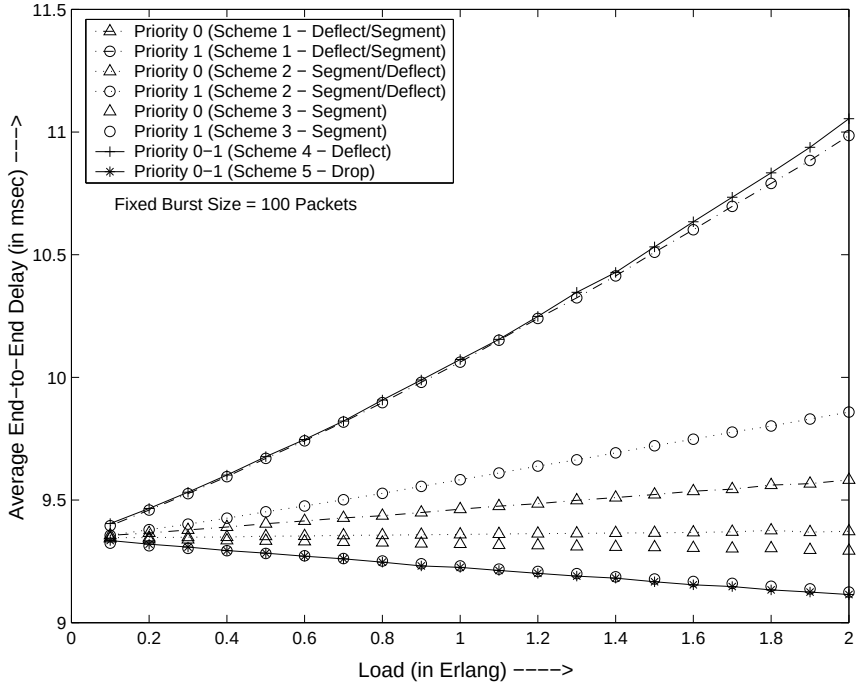


Figure 5.5. Average end-to-end packet delay versus load for different QoS schemes with fixed burst size = 100 packets, with the traffic ratio being 20% Priority 0 and 80% Priority 1 bursts.

graph shows packet losses for the case in which 20% of the traffic is high-priority and 80% of the traffic is low-priority. We observe that the loss of high-priority packets are lower than that for low priority packets in schemes which employ burst segmentation (Scheme 1, 2, and 3), while schemes without segmentation do not provide service differentiation (Scheme 4 and 5). We also observe that Scheme 1 performs the best under the observed load values, while Scheme 2 performs better at higher loads; thus, at low loads, it is better to attempt deflection before segmentation when two bursts are of equal priority. At higher loads, schemes with deflection as the primary contention resolution technique (Scheme 1 and 4) suffer from higher loss compared to schemes with no or controlled deflection (Scheme 2 and 3) due to the increased load due to deflection. Also, by varying the number of alternate deflection ports at each switch, we can achieve different levels of packet loss.

Figure 5.5 plots the average end-to-end packet delay versus load for high-priority and low-priority packets for Scheme 1 through Scheme 5, with fixed-sized bursts. We observe that the delay of high-priority packets are lower than that for low-priority packets in schemes which employ burst segmentation (Scheme 1, 2, and 3). Schemes without segmentation do not provide service differentiation (Scheme 4 and 5), and hence have the same delays for both priorities. The delay for high-priority bursts remains in a consistent range, while the low-priority bursts have higher delay due to multiple deflections. At very high load, bursts which are further from their destination are less likely to reach their destination compared to those bursts which are close to their destination; thus, the average delay will eventually decrease at very high load. Schemes 1 and 4 suffer high delays compared to other schemes, since the contending burst (either lower or equal-priority) is deflected first.

In order to evaluate the performance of the segmentation and deflection schemes, we develop a simulation model. The following have been assumed to obtain the results:

- Burst arrivals to the network are Poisson with rate λ.
- Burst length is exponentially distributed with average burst length of $1/\mu$ = 100 ms.

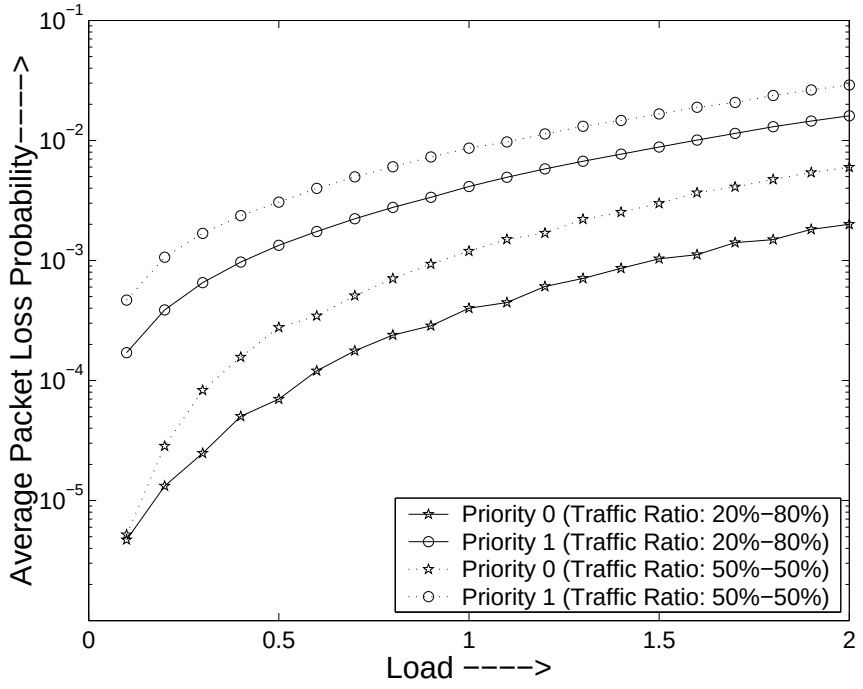


Figure 5.6. Packet loss probability versus load for different traffic ratios using Scheme 1.

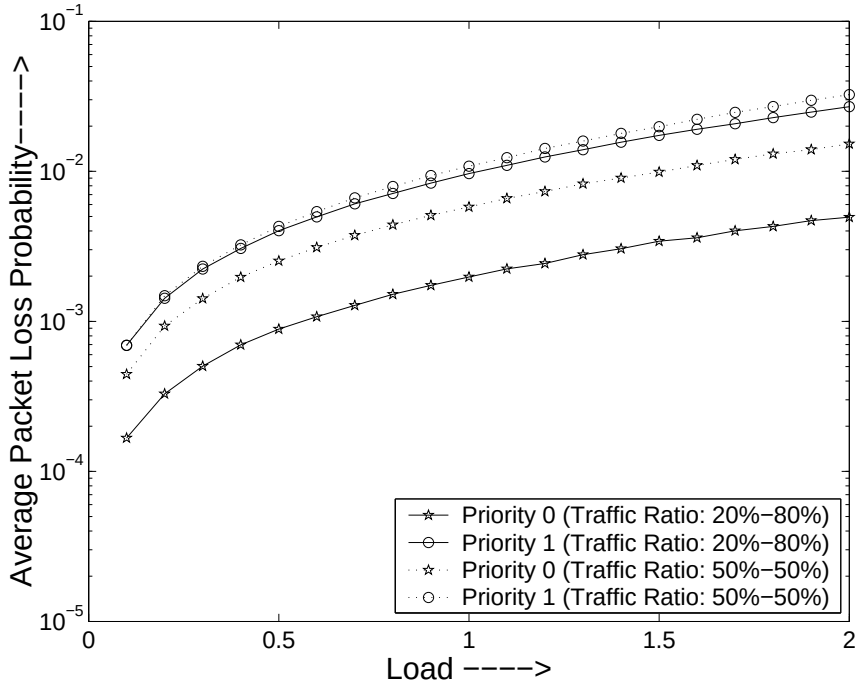


Figure 5.7. Packet loss probability versus load for different traffic ratios using Scheme 2.

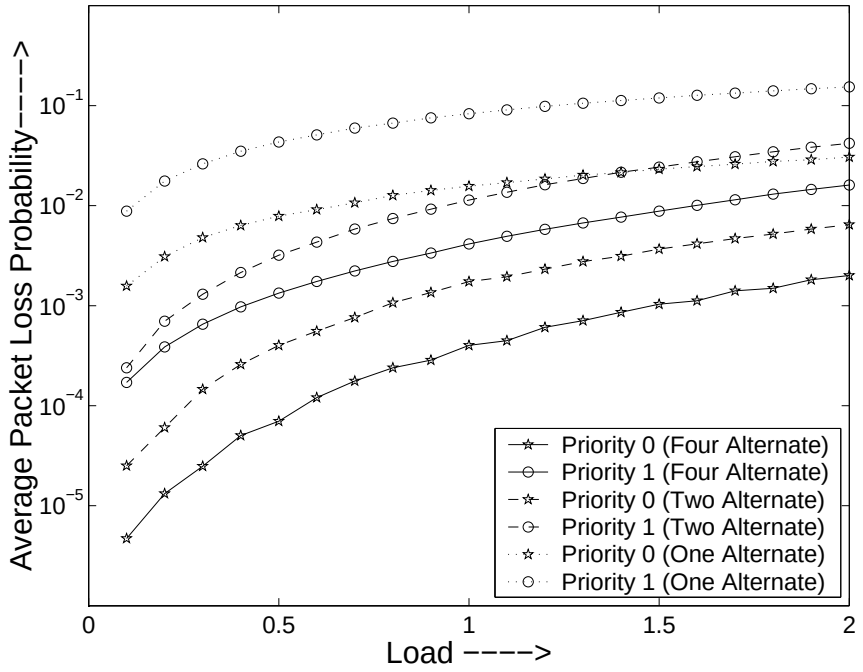


Figure 5.8. Packet loss probability versus load Scheme 1 with different number of alternate deflection ports.

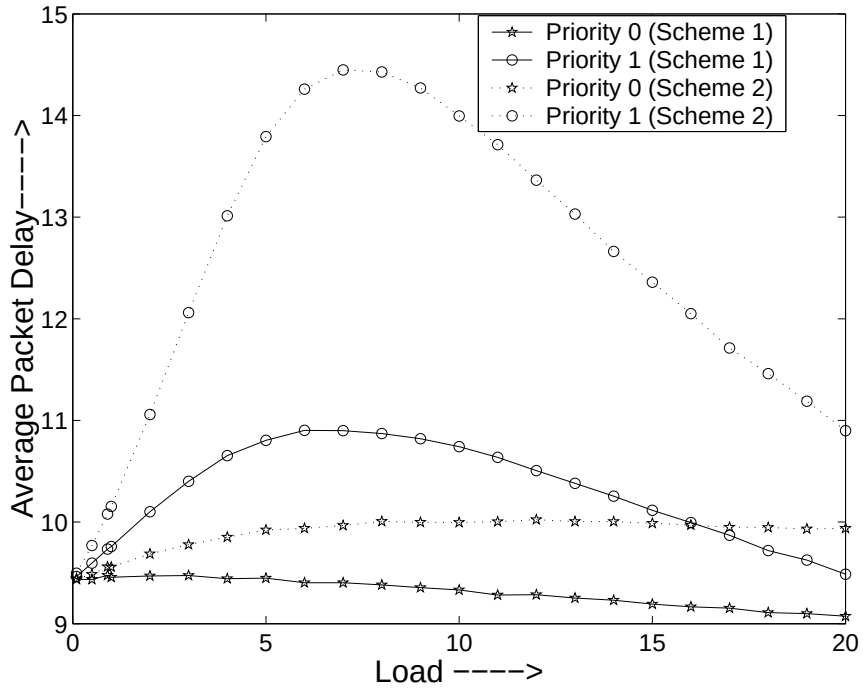


Figure 5.9. Average packet delay versus load using Scheme 1 and Scheme 2.

- Load is measured in Erlang.
- 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.
- Each node handles both bypassing and locally generated or terminated bursts.
- Bursts are uniformly distributed over all sender-receiver pairs.
- Dijkstra shortest path routing algorithm is used to find the path between all node pairs.

Figures 5.6 and 5.7 plot the packet loss probability versus load for high-priority (Priority 0) and low-priority (Priority 1) packets, using Scheme 1 and Scheme 2 respectively. Each shows packet losses for the case in which there is an equal amount of high-priority and low-priority traffic, and the case in which 20% of the traffic is high priority and 80% of the traffic is low priority. We observe that the loss of high-priority packets is lower than that for low priority packets. We also observe that Scheme 1 performs better than Scheme 2 at these loads; thus, at low loads, it is better to attempt deflection before segmentation when two bursts are of equal priority.

Figure 5.8 plots total packet loss probability versus load for different number of alternate deflection ports with 20% of high-priority and 80% of low-priority traffic. We observe that there is a significant improvement when we use two alternate deflection ports instead of one alternate port, while there is less improvement from two to four alternate deflection ports. This result is due to the low nodal degree of NSF network (Figure 5.2) and may differ for other networks.

Figure 5.9 plots total delay versus load with 20% of high-priority and 80% of low-priority traffic for the two QoS schemes. Scheme 2 has lower delays compared to Scheme

1, as Scheme 2 follows the segment-first approach rather than the deflect-first approach. The delay for high-priority bursts remains in a consistent range, while the low-priority bursts have higher delay due to multiple deflections. At very high load, bursts which are farther from their destination are less likely to reach their destination compared to those bursts which are close to their destination; thus, the average delay will eventually decrease as load increases.

5.5 Conclusion

In this chapter, we introduce the concept of prioritized contention resolution through prioritized burst segmentation and deflection to provide QoS in the optical burst-switched core network. The prioritized contention resolution policies can provide QoS with 100% class isolation without requiring any additional offset times. An analytical model for prioritized burst segmentation was developed to calculate the packet loss probabilities for a two-priority network, and the model was verified through simulation. The high-priority bursts have significantly lower losses and delay then the low-priority bursts, and the schemes which incorporate deflection tend to perform better than the schemes with limited deflection or no deflection. Also, prioritized burst segmentation is easily scalable in order to support multiple priorities in an all-optical burst-switched network.

CHAPTER 6

COMPOSITE BURST ASSEMBLY TECHNIQUES FOR PROVIDING QOS SUPPORT IN OPTICAL BURST-SWITCHED NETWORKS

6.1 Introduction

An important issue in optical burst switching is burst assembly. Burst assembly is the process of aggregating and assembling IP packets into a burst at the edge of the network. The most common burst assembly approaches are *timer-based* and *threshold-based*. In a timer-based burst assembly approach, a burst is created and sent into the optical network at periodic time intervals [43]; hence, the network may have variable length input bursts. In a threshold-based approach, a limit is placed on the number of packets contained in each burst; hence, the network will have fixed-size input bursts [150]. Timer-based and threshold-based approaches may also be combined into a single burst assembly scheme.

In this chapter, we focus on the issue of providing QoS support in OBS through prioritized burst segmentation (Ch. 5) and composite burst assembly. In the prioritized contention resolution scheme, priorities are included as a field in the BHP. This priority field is used to preferentially segment and deflect bursts when resolving contentions in the core. The composite burst assembly technique is implemented at the OBS network edge and assembles packets of different IP packet classes into the same burst in an attempt to meet the delay and loss constraints of each IP packet class. We develop a generalized framework for describing a wide range of burst assembly schemes and provide specific examples of composite burst assembly schemes. Analytical and simulation models are developed to evaluate the packet loss probability of the various QoS schemes. In this work, we assume that JET signaling is used and that there are no fiber delay lines or wavelength converters in the network. The QoS re-

quirements of an IP packet are defined by the packet's *Class*, whereas bursts are differentiated in the core based on assigned *Priorities*.

The remainder of the chapter is organized as follows. Section 6.2 describes the generalized burst assembly framework. Section 6.3 describes the proposed burst assembly techniques. In Section 6.5, we develop an analytical model to calculate the packet loss probability for the proposed composite burst assembly. Section 6.6 provides numerical results from simulation and analysis and compares the results of the different burst assembly schemes. Section 6.7 concludes the chapter.

6.2 Generalized Burst Assembly Framework

In this section, we formulate a generalized framework for burst assembly. The primary issues are which class of packets and how many of packets of each QoS class to put into a burst. To provide QoS support, the burst assembly policies should take into account the number of packet classes as well as the number of burst priorities supported in the core. A burst can contain packets of a particular class (Fig. 6.2(a)), or a combination of packets of different classes (Fig. 6.2(b)). Existing burst assembly techniques assemble packets of the same class into a burst. We introduce a new approach of assembling packets of different classes into a single burst, namely, *composite burst assembly*. This approach is motivated by the observation that, with burst segmentation, if we consider the strict tail-dropping approach, the packets toward the tail of a burst are more likely to be dropped than packets at the head of a burst; thus, packet classes which have low loss tolerance may be placed toward the head of a burst while packet classes which have higher loss tolerance may be placed toward the tail of a burst. Note that the reverse ordering of packets inside a burst would apply if a strict head-dropping approach is adopted in the OBS core. If both tail-dropping and head-dropping is used, such as in the case of non-preemptive burst segmentation (Chapter 6.4), then different packet-ordering schemes need to be considered during burst assembly. In this chapter, we only consider the tail-dropping approach, since it facilitates prioritized contention resolution

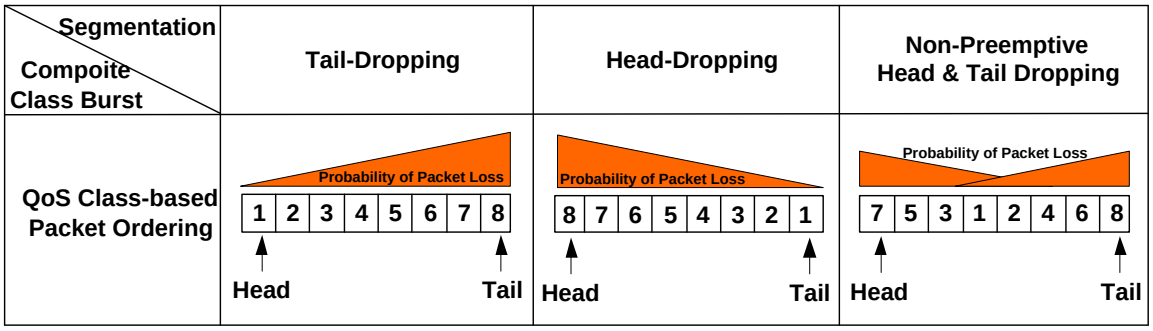


Figure 6.1. Different Composite Class Bursts based on the supported burst segmentation policies in the core; (a) for strict tail-dropping, (b) for strict head-dropping, and (c) for non-preemptive (both head-dropping and tail-dropping).

in the bufferless OBS core. Figure 6.1 provides an illustrative example of possible packet ordering in a composite burst assembly technique with the three different burst segmentation techniques, namely tail-dropping (Fig. 6.1(a)), head-dropping (Fig. 6.1(b)), non-preemptive (Fig. 6.1(c)). By implementing composite burst assembly, the network can support differentiation even if the number of IP packet classes exceeds the number of burst priorities supported in the core.

Another issue in burst assembly is when to create a burst. Typically, threshold and timer based approaches are used to determine when a burst should be created. In a timer-based approach, a timer is started when a packet arrives. When the timer expires, a burst is created from all packets received. In a threshold-based approach, an upper bound is placed on the number of packets in the burst. When the threshold is reached, a burst is created. Below, we provide a generalized framework for classifying various burst assembly approaches.

Let N be the number of input packet classes at the edge and let M be the number of burst priorities supported in the core network. Given N packet classes and M burst priorities, the objective is to meet the QoS requirements by defining a set of *burst types* which specify how packets are aggregated, and by assigning an appropriate burst priority to each burst type. In this model, we define the length of the burst by the number of packets in the burst. Let K be the number of burst types, where $M \leq K \leq (2^N - 1)$. A burst type of type k 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.
- P_k: priority of burst of type k.
- τ_k: timeout value for creating bursts of type k.
- T_k: threshold value for creating bursts of type k.
- C_k: $C_k \subseteq S_k$, subset of packet classes over which the threshold is evaluated. If x_j is defined as the number of packets of Class j at the ingress node, then a burst is created if $\sum_{j \in C_k} x_j \geq T_k$.

The burst creation criterion for a burst of type k is satisfied either when the threshold value T_k for packets in C_k is satisfied, or when the timeout value, τ_k is reached. When the criterion is satisfied, a burst of type k is created, and the classes of packets to be included in the burst are specified by S_k. Packets are added to the burst until L_k^{MAX} is reached.

For example, in a threshold-based approach ($T_k \geq L_k^{MIN}$), if $S_k = \{1, 2\}$, then C_k can be {1,2}, {1}, or {2}. If C_k = {1,2}, then a burst of type k is created when the sum of packets of Class 1 and Class 2 is $\geq T_k$. If C_k = {1}, then a burst of type k is created when the number of packets of Class 1 is $\geq T_k$. If C_k = {2}, then a burst of type k is created when the number of packets of Class 2 is $\geq T_k$. In each of these cases, packets of both Class 1 and Class 2 may be included in the burst until L_k^{MAX} is reached.

6.3 Burst Assembly Techniques

We now provide general guidelines for defining various burst types. The important design considerations when defining the burst types are packet loss probability, delay constraints, and bandwidth requirements. By appropriately mapping packet classes to burst types and by assigning appropriate priorities, P_k, to burst types, differentiated levels of packet loss may be achieved. End-to-end delay constraints can be met by setting appropriate timeout values, τ_k for each burst type. Bandwidth requirements can be met by choosing an appropriate R_{jk}^{MIN} and R_{jk}^{MAX} for each packet class. In this chapter, we focus primarily on achieving differentiated loss and delay. A fixed value of T_k, is assigned for all burst types, and a timeout value, τ_k is assigned only to the highest priority burst. We investigate the following approaches for selecting mappings S_k and priorities P_k to achieve differentiated QoS.

6.3.1 Approach 1: Single Class Burst (SCB) with $N = M$

For the case in which $N = M$, we can create $K = M$ burst types such that each burst only contains a single class of packets ($S_k = \{k\}$). The priority of a burst will be equal to the class of packets contained in the burst ($P_k = k$). If a threshold based approach is adopted, then the threshold, T_k for a Priority k burst will be evaluated over Class k packets ($C_k = \{k\}$).

For example, if $N = 4$ and $M = 4$, as shown in Fig. 6.2(a), we set the number of burst types, K, equal to 4. We set $S_0 = C_0 = \{0\}$, $S_1 = C_1 = \{1\}$, $S_2 = C_2 = \{2\}$, and $S_3 = C_3 = \{3\}$. If we consider the Class 2 packets that are collected in an input queue, once the number of Class 2 packets exceeds T_2, a burst consisting of Class 2 packets is created and sent into the network with a burst Priority 2. This process is followed for each class; thus, the priority of a burst will directly correspond to a specific class of packets contained in the burst.

6.3.2 Approach 2: Composite Class Burst (CCB) with $N = M$

In composite bursts, each burst can consist of packets of different classes. One approach is to have $K = M$ burst types with a burst of type k containing packets of both Class k and Class

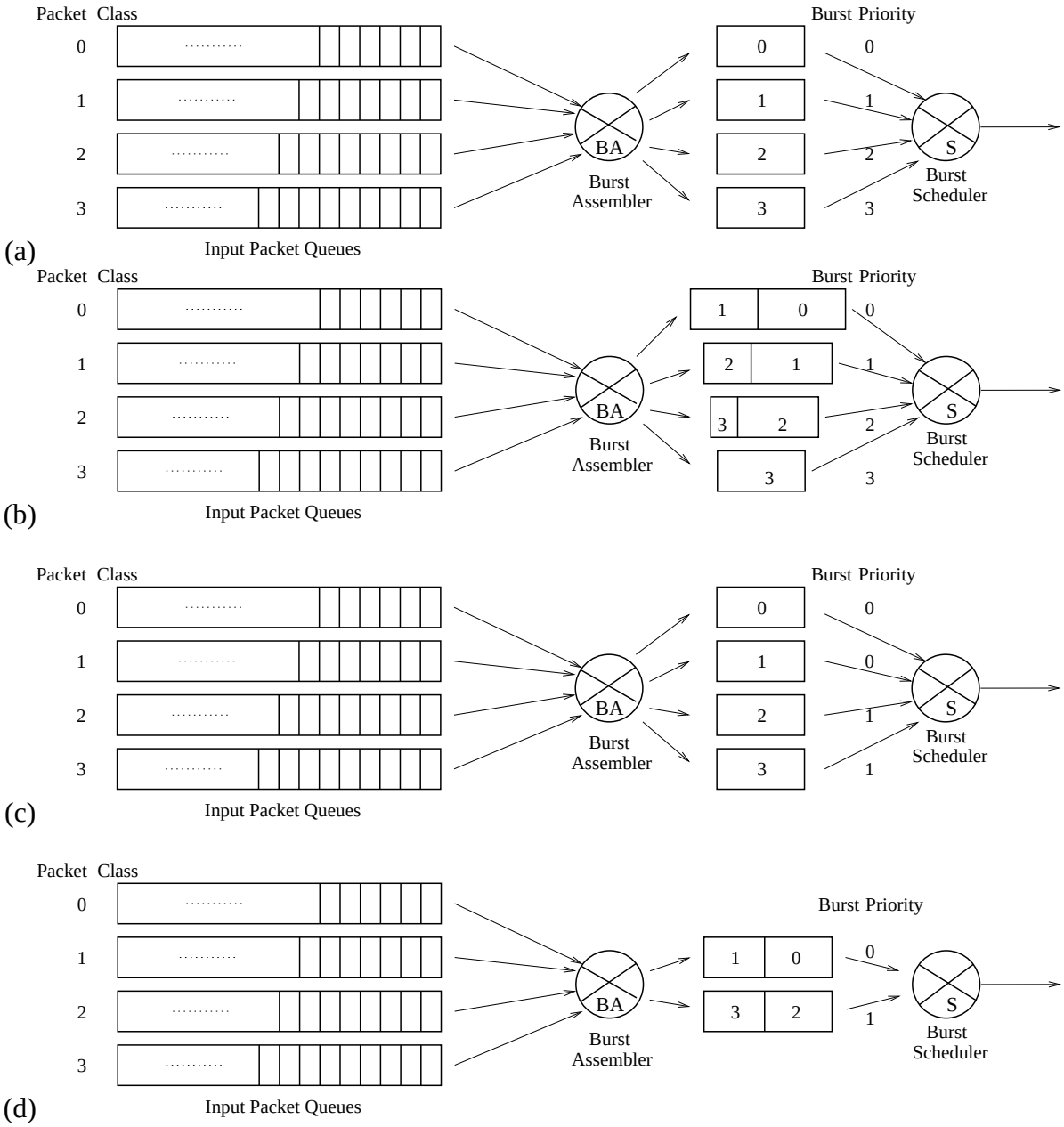


Figure 6.2. (a) Creation of Single Class Burst with N = 4 and M = 4. (b) Creation of Composite Class Burst with N = 4 and M = 4. (c) Creation of Single Class Burst with N = 4 and M = 2. (d) Creation of Composite Class Burst with N = 4 and M = 2.

$k+1$, i.e., $S_k = \{k, k+1\}$. In this approach, packets are placed in the burst in decreasing order of class, such that the higher class packets are at the head of the burst. A burst of type k is generated if the number of packets of Class k is equal to the threshold T_k ($C_k = \{k\}$) or if the timeout τ_k has expired. The priority of the burst is given by the burst type ($P_k = k$).

For example, if $N = M = 4$, as shown in Fig. 6.2(b), we set the number of burst types, K, equal to 4, and we also set the following parameters: $S_0 = \{0, 1\}, S_1 = \{1, 2\}, S_2 = \{2, 3\}$, $S_3 = \{3\}$, C_0 = {0}, C_1 = {1}, C_2 = {2}, and C_3 = {3}. If the threshold of packet Class 1 is met, then a burst of type 1 is created with packets of class S_1 = {1,2}, where Class 1 packets are placed at the head of the burst and Class 2 packets are placed at the tail of the burst. It is important to notice that there is no additional overhead incurred when ordering packets during the creation of the burst, since it is possible to access a particular input packet queue, place its contents in a burst, then go to the next lower class queue. This process can be repeated for all packet classes in S_k.

In the case of a contention, burst priorities are compared. If the priorities are equal, the tail of the original burst is dropped. Dropping the tail of the original burst effectively gives the tail of a burst lower priority then the head of a burst. In such a scheme, during a contention of equal-priority bursts, lower class packets are dropped for the benefit of the higher class packets.

6.3.3 Approach 3: Single Class Burst (SCB) with $N > M$

We now consider single-class bursts for the case $N > M$. In this approach we have $K = N$ types of bursts, where each burst consists of packets of a single class ($S_k = \{k\}$). However, several burst types will have the same burst priority given by, $P_k = \lfloor kM/N \rfloor$.

For example, if $N = 4$ and $M = 2$, as shown in Fig. 6.2(c), we set the number of burst types, K, equal to 4. We have four unique types of bursts, each containing a single class of packets, i.e., $S_0 = C_0 = \{0\}$, $S_1 = C_1 = \{1\}$, $S_2 = C_2 = \{2\}$, and $S_3 = C_3 = \{3\}$. Each burst is assigned one of the two burst priorities. Bursts containing either Class 0 or Class 1

packets have Priority 0, while bursts containing either Class 2 or Class 3 packets have Priority 1.

6.3.4 Approach 4: Composite Class Burst (CCB) with $N > M$

We now consider composite-class bursts for the case $N > M$. In this case we have $K = M$ types of burst, where each burst consists of packets of class given by, $S_k = \{\frac{kN}{M}, ..., \frac{(k+1)N}{M} - 1\}$. A burst of type k is generated if the sum of packets of classes in C_k is equal to the threshold T_k ($C_k = S_k$). Once the threshold or timer criterion is met, a burst of type k containing packets defined by S_k is generated by appending all constituent class packets into the burst in decreasing order of class, such that the highest class packet in that burst type is at the head of the burst. The priority of the burst is same as type of burst ($P_k = k$).

For example, if $N = 4$ and $M = 2$, as shown in Fig. 6.2(d), we set the number of burst types, K, is equal to 2. We select $S_0 = C_0 = \{0, 1\}$ and $S_1 = C_1 = \{2, 3\}$. If the sum of Class 0 and Class 1 packets meet the threshold T_0, then a burst of type 0 is created with packets of class $S_1 = \{0, 1\}$. The two types of composite bursts $\{0, 1\}$ and $\{2, 3\}$ are assigned burst Priority 0 and Priority 1 respectively.

6.4 Burst Scheduling Techniques

Once a burst is created it must be sent into the OBS core. Burst scheduling is the problem of sending the created bursts into the core such that the loss, delay, and bandwidth constraints of each class are met.

Burst Scheduling for supporting QoS in OBS networks is different from traditional IP scheduling disciplines. In IP, each core node stores the packets in prioritized buffers and schedules them. In OBS, we must consider the scheduling of electronically buffered burst at the ingress, while simultaneously handling the all-optical transit traffic. Hence, in case of a contention at the source, where the intended output port has been occupied by a transit burst of priority P_x, the burst scheduling policy has to take into account the relative priorities of each

new burst versus P_x. The different edge scheduling techniques are described in Section 2.5.

In this chapter, the created bursts are sent in FCFS order. In case the outgoing port is occupied by a transit burst, the burst priorities are compared. If the created burst has higher priority than the transit burst, then it preempts the transit burst. In the above examples, we assumed a simple first-come-first-serve scheduling policy; however, in order to achieve greater control over the delay and bandwidth metrics, it may be desirable to implement more intelligent burst scheduling policies.

6.5 Analytical Model

We now compute the packet loss probability for different packet classes in a composite class burst (CCB). We consider an OBS network with four packet classes and two burst priorities. Let Class 0, Class 1, Class 2, and Class 3 be the four packet classes with Class 0 being the highest packet class and Class 3 being the lowest packet class, in that order. Let Priority 0 and Priority 1 be the high-priority and low-priority bursts supported in the networks.

The following are the assumptions:

- Initial burst length is fixed.
- T^0: high-priority burst length.
- T^1: low-priority burst length.
- α: ratio of Class 0 packets in the high-priority burst.
- β: ratio of Class 2 packets in the low-priority burst.
- The ratio of traffic of Class 1 and Class 4 will be $(1-\alpha)$ and $(1-\beta)$ in the high and low-priority bursts respectively.
- Class 0 packets are placed towards the head and Class 1 packets are placed towards the tail of the high-priority burst.

- Class 2 packets are placed towards the head and Class 3 packets are placed towards the tail of the low-priority burst.

From Section 5.3, we can then find the average packet loss probability of high-priority and low-priority bursts for the system by finding the individual loss probability for each source-destination pair, and taking the weighted average of the loss probabilities:

$$P_{loss}^{0} = \sum_{s} \sum_{d} \frac{\lambda^{sd}}{\lambda} P_{loss}^{sd}. \tag{6.1}$$

$$P_{loss}^{1} = \sum_{s} \sum_{d} \frac{\gamma^{sd}}{\gamma} P_{loss1}^{sd}. \tag{6.2}$$

Based on the ratio of packets of each class, we can find the packet loss probabilities of each class. The packet loss for Class 0, P_{loss}^{00}, is the same as the loss probability of a high-priority burst of length $\alpha \cdot T^0$; therefore, we can obtain P_{loss}^{00} by replacing T^0 in (5.8) with $\alpha \cdot T^0$. The packet loss probability for Class 1 is found by considering the total packet loss probability in a burst and the packet loss probability of Class 0 packets; thus, P_{loss}^{01} is given by:

$$P_{loss}^{01} = \frac{P_{loss}^{0} - \alpha \cdot P_{loss}^{00}}{1 \; - \; \alpha}. \tag{6.3}$$

Similarly, the packet loss probability for Class 2, P_{loss}^{12}, is same as the packet loss probability of a low-priority burst of length $\beta \cdot T^1$, and can be found by replacing T^1 in (5.22) with $\beta \cdot T^1$. The packet loss probability for Class 3 is given by:

$$P_{loss}^{13} = \frac{P_{loss}^{1} - \beta \cdot P_{loss}^{12}}{1 \; - \; \beta}. \tag{6.4}$$

6.6 Numerical Results

In order to evaluate the performance of the proposed schemes and to verify the analytical models, a simulation model is developed. Burst arrivals to the network are assumed to be Poisson with rate λ. Burst lengths are exponentially distributed with average length of $1/\mu$ =

100 ms. The link transmission rate is 10 Gb/s. Packets are assumed to be 1250 bytes, and each segment consists of a single packet. The configuration time of the switching is assumed to be 10 μs. There is no buffering or wavelength conversion at the core nodes. Burst arrivals are uniformly distributed over all sender-receiver pairs, and shortest-path routing is assumed. Figure 6.3 shows the 14-node NSF network on which the simulation was implemented.

6.6.1 Analytical Results

Let us consider a network with two priorities. The fraction of high-priority (Priority 0) bursts is 20%, and the fraction of low-priority (Priority 1) bursts is 80%. In the analytical model, we ignore the switching time and header processing time.

We consider an OBS network with composite bursts. The network supports four packet classes. Class 0 is the highest packet class and Class 3 is the lowest packet class. Fig. 6.4 plots the packet loss probability versus α and β for Scheme 3 without length comparison, where α is the ratio of Class 0 packets in the high-priority burst, and β is the ratio of Class 2 packets in the low-priority burst. The graphs are plotted for a fixed load of 1 Erlang with fixed-sized bursts. We observe that the packet loss probability of the different classes obtained through the analytical models match with the simulation results. Also, the analytical model slightly over-estimates the packet loss probabilities due to the independent link assumption. By choosing a specific value of α and β, we can ensure that a certain level of performance is guaranteed. For example, for the case shown in Fig. 6.4, if we choose $\alpha = 55\%$, then the packet loss probability of Class 0 will be less than 1%.

6.6.2 Simulation Results

We consider composite and single burst assembly while utilizing Scheme 3 without length comparisons for contention resolution in the core. The input traffic ratios of individual packet classes are 10%, 20%, 30%, and 40% for Class 0, Class 1, Class 2, and Class 3 respectively. We set a threshold value of 100 packets for each burst type, and a timeout value of 50 ms for

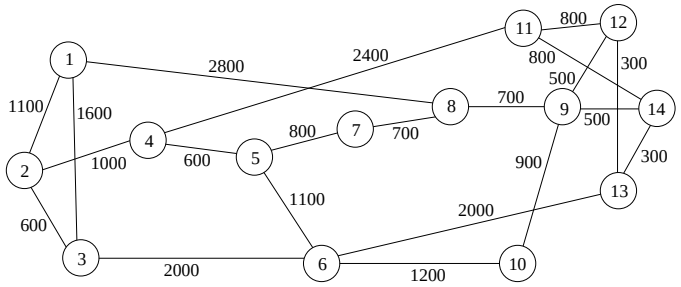


Figure 6.3. NSF network with 14 nodes (distances in km).

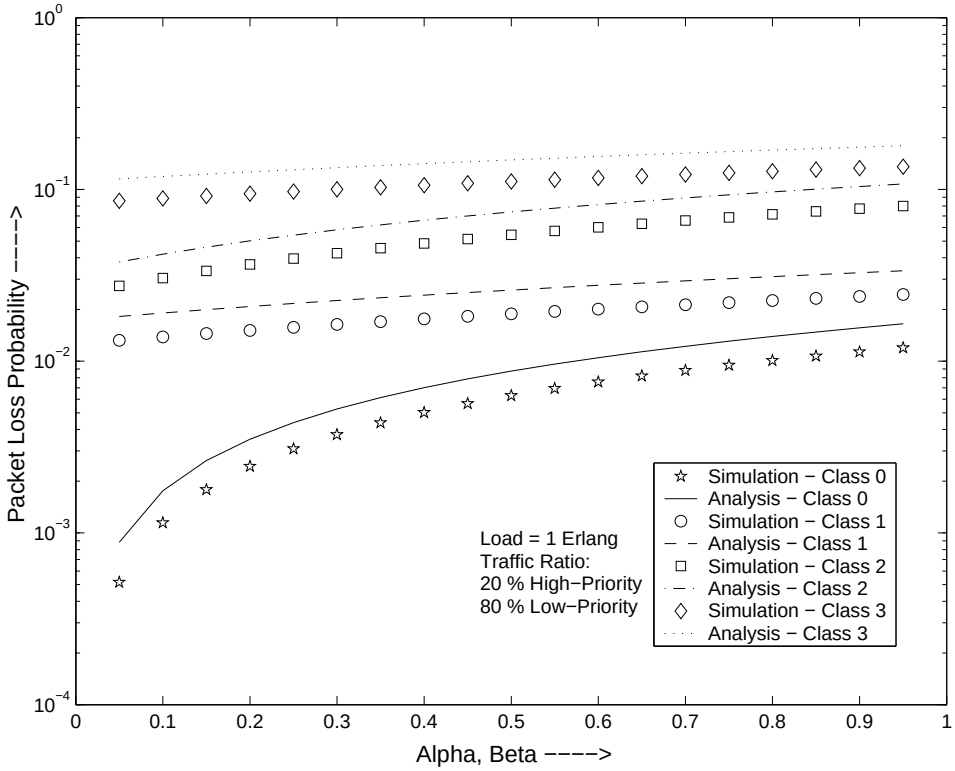


Figure 6.4. Packet loss probability versus alpha and beta values for composite bursts of fixed initial burst size = 100 packets length using Scheme 3 without length comparison.

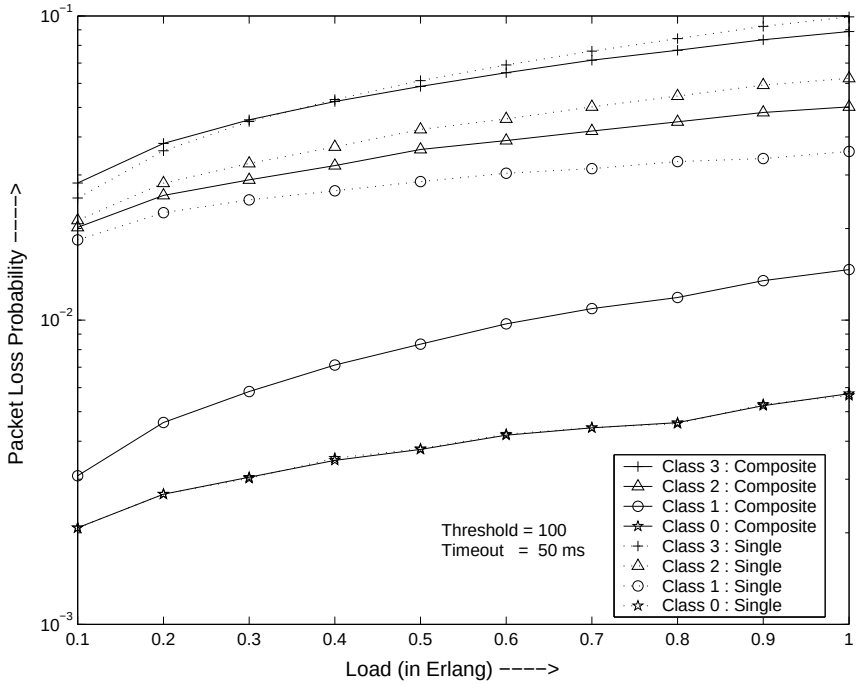


Figure 6.5. Packet loss probability versus load for N = 4 and M = 4 for single and composite class bursts, with the traffic ratio of the packets classes A, B, C, D being 10%, 20%, 30%, 40%.

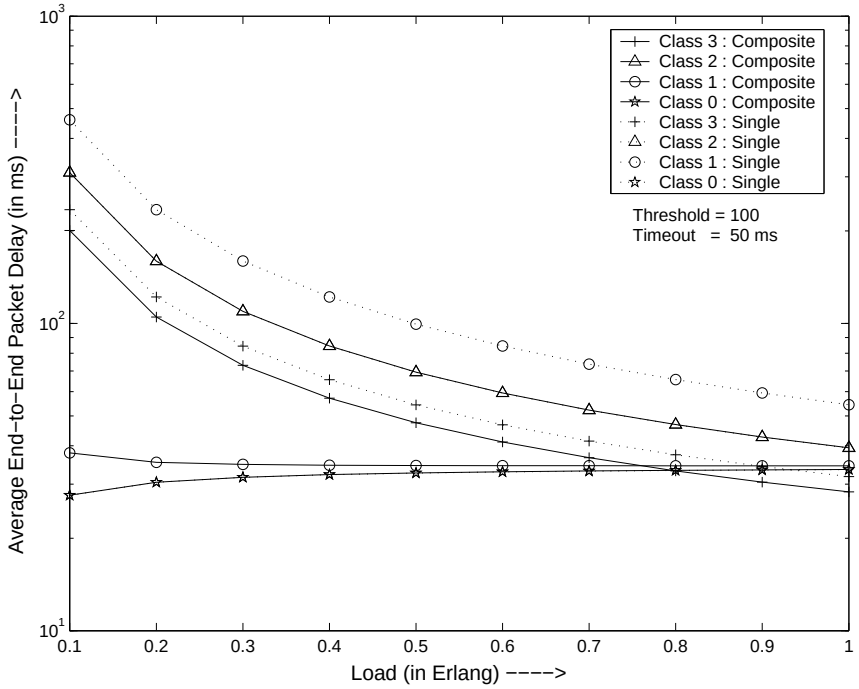


Figure 6.6. Average End-to-End packet delay versus load for N = 4 and M = 4 for single and composite class bursts, with the traffic ratio of the packets classes A, B, C, D being 10%, 20%, 30%, 40%.

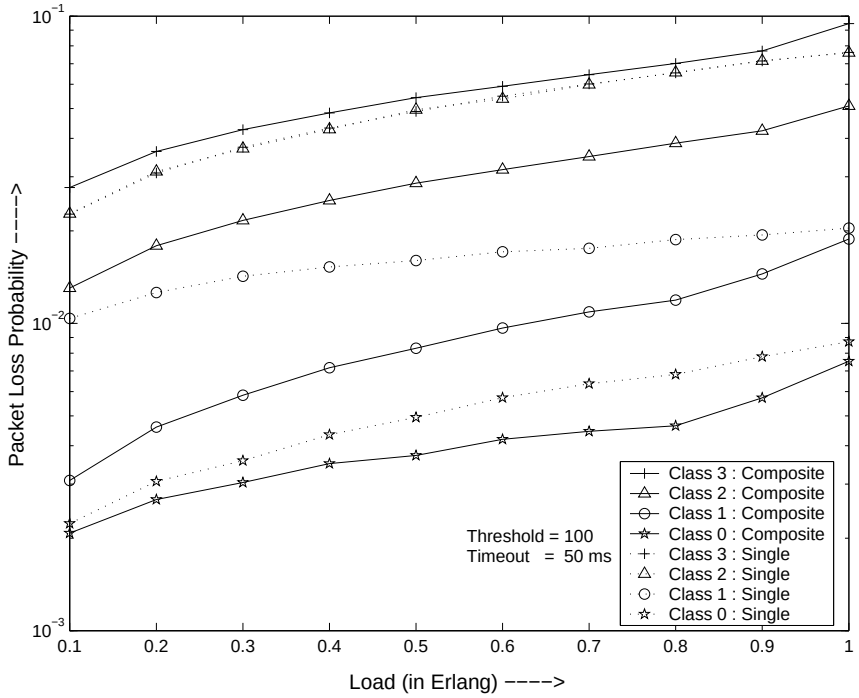


Figure 6.7. Packet loss probability versus load for N = 4 and M = 2 for single and composite class bursts, with the traffic ratio of the packets classes A, B, C, D being 10%, 20%, 30%, 40%.

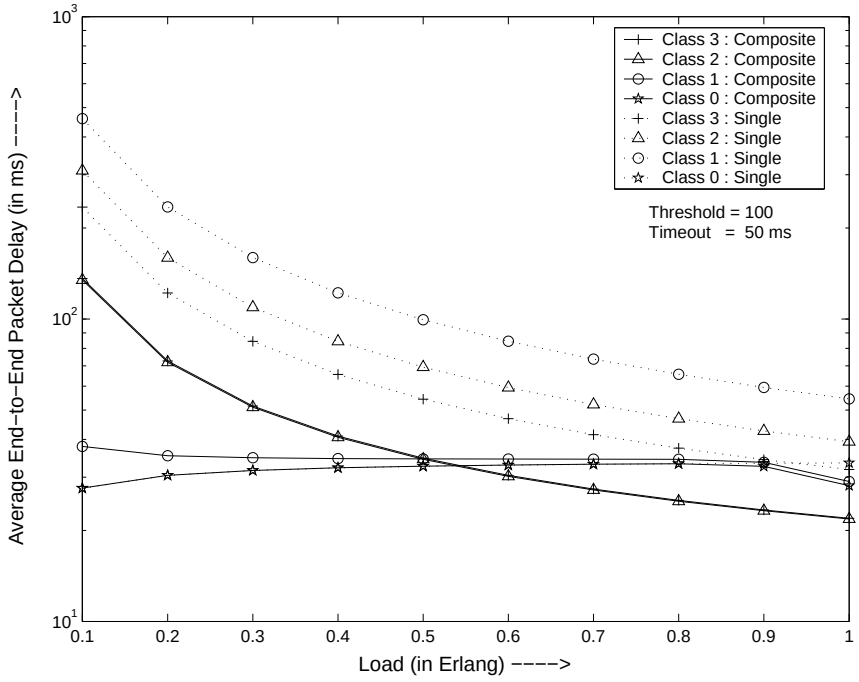


Figure 6.8. Average End-to-End packet delay versus load for N = 4 and M = 2 for single and composite class bursts, with the traffic ratio of the packets classes A, B, C, D being 10%, 20%, 30%, 40%.

the highest priority burst. We also avoid contentions between multiple bursts at the source by delaying the contending bursts until the desired output port is free. The remaining assumptions remain the same as the prioritized burst segmentation case.

Figures 6.5 and 6.6 plot packet loss probability and average end-to-end delay versus load for both CCB and SCB with $N = M = 4$. We refer to this case as the 4:4 mapping. We observe that, by using CCB, the loss of packets is more proportional to the packet class than in SCB. We observe that the loss of lower class packets is better in CCB, since some of the lower class packets are placed into higher priority bursts, which, in turn, decreases the loss probability. Also, the highest class packets in CCB perform as well as in SCB, since, at every contention between highest priority bursts, the lower-class packets are more likely to be dropped. We see that the average delay decreases with the increase in load. This decrease is due to the higher arrival rate of packets which causes the threshold to be satisfied more frequently. The delay of highest class packets is fairly constant, since we enforce an upper-limit on the aggregation time by using a timeout.

Figures 6.7 and 6.8 plot packet loss probability and average end-to-end delay versus load for both CCB and SCB with $N = 4$ and $M = 2$. We refer to this case as the 4:2 mapping. We observe that the performance of CCB is much better than SCB for the highest class packets. This is due to the fact that in a 4:2 mapping, both packets of Class 0 and Class 1 are assigned Priority 0, and in an equal-priority contention, packets of Class 1 may preempt packets of Class 0. In SCB, the loss of Class 0 packets and Class 1 packets will be the same if the input ratio are the same, and if the same threshold and timeout values are used. In our example, a timeout value is assigned to bursts carrying Class 0 packets, but not to bursts carrying Class 1 packets. This difference results in lower loss and delay for Class 0 packets, even though the burst are of equal-priority. Also, we see that the average end-to-end delay for Class 0 and Class 1 in the case of CCB are similar in both 4:4 and 4:2 mapping, since Class 1 packets are included in the same bursts as Class 0 packets when the timeout is reached. The difference in delay between Class 0 and Class 1 packets is due to their different arrival rates.

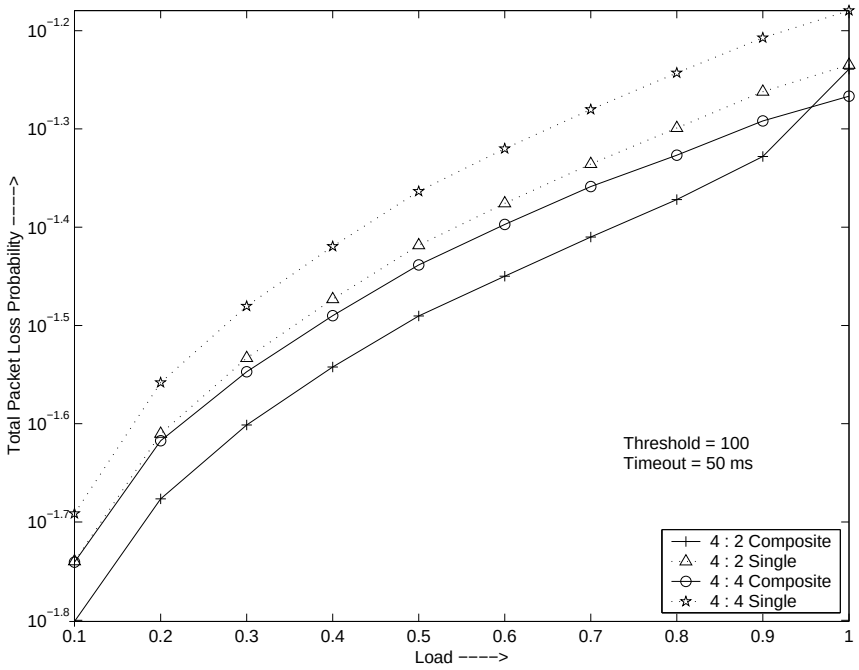


Figure 6.9. Packet loss probability plotted versus load.

Figure 6.9 plots total packet loss probability versus load for both CCB and SCB for both of the above cases respectively. We see that the total loss using the CCB technique is much lower than when using SCB. This is due to the reduction in the number of different priority bursts, which leads to increased probability of equal-priority contentions. Also, we see that 4:2 mapping has lower loss than 4:4 mapping, since the number of contentions between different priority bursts is lower in 4:2 mapping as compared to 4:4 mapping. According to the segmentation policy, more packets are dropped in a contention involving burst of different priority as compared to contention of the same priority.

6.7 Conclusion

We introduced the concept of composite burst assembly to handle the differentiated service requirements of the IP packets at edge nodes of the optical burst-switched network, and we described a generalized framework for burst assembly.We considered four different burst as-

sembly approaches and evaluated their performance in terms of delay and loss. We observe that approaches with composite bursts perform better than approaches with single-class bursts with respect to providing differentiated QoS for different classes of packets. This was verified by the analytical model results. The developed model can be useful for selecting the class ratios for composite bursts in a manner which can satisfy the packet loss requirement. In order to further reduce the packet loss, the proposed techniques can be employed in conjunction with all-optical wavelength conversion and buffering through fiber delay lines.

CHAPTER 7

THRESHOLD-BASED BURST ASSEMBLY POLICIES FOR PROVIDING QOS SUPPORT IN OPTICAL BURST-SWITCHED NETWORKS

7.1 Introduction

Burst assembly is the process of aggregating and assembling input packets into bursts at the edge of the OBS network. 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 [33]; hence, the network may have variable length input bursts. 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.

In burst assembly, a significant issue is how to decide on the appropriate burst length for specific network parameters in order to minimize the packet loss probability in the OBS network. We can clearly observe that, for a given amount of data, creating longer bursts will reduce the total number of bursts injected into the OBS network; however, in the case of a contention, the average number of packets lost per contention will increase. On the other hand, generating smaller bursts will increase the number of bursts in the OBS network, leading to a greater number of contentions, and therefore higher packet loss probability. Thus, there exists a tradeoff between the number of contentions and the average number of packets lost per contention, and it is expected that the performance of an OBS network can be improved if the incoming packets are assembled into bursts of optimal length.

One of the major reasons for data loss in OBS networks is burst contention, which occurs when multiple bursts contend for the same output link. Contention in an OBS network is particularly aggravated by the highly variable burst sizes (Figure 3.7) and the long burst durations. Packet losses due to contention can be reduced through *burst segmentation* [106]. Burst segmentation is a process in which only those parts of a burst which overlap with another burst are dropped.

In this chapter, we investigate threshold-based burst assembly techniques and their effect on the packet loss performance in an optical burst-switched network. We also study the effect of burst assembly on providing QoS support in an OBS network.

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. Our 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, we assume 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.

We consider an OBS network which uses the JET signaling technique with burst segmentation. 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 .

The chapter is organized as follows. Section 7.2, discusses the architecture of burst assembler at the OBS edge node. In Section 7.3, we discuss threshold-based approaches used for providing QoS. In Section 7.4, we provide the simulation results and show how different threshold-based approaches provide QoS support in the network. We conclude the chapter in Section 7.5.

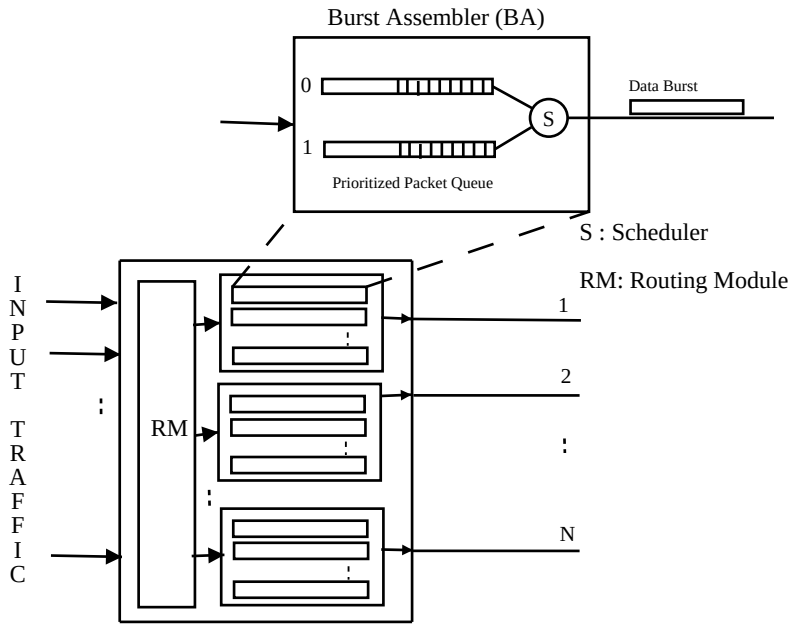


Figure 7.1. Architecture of Edge Node with Burst Assembler.

7.2 Edge Node Architecture

An OBS network consists of a collection of edge and core routers. The edge routers shown in Figure 7.1, assemble the electronic input packets into an optical burst which is sent over the OBS core. The ingress node pre-sorts and schedules the incoming packets into electronic input buffers according to each packet's class and destination address. The packets are then aggregated into bursts that are stored in the output buffer. Since a separate buffer is required for each packet class and each destination, the limit on the maximum number of supported packet class is determined by the maximum electronic packet buffer size at each ingress node. The assembled bursts are transmitted all-optically over OBS core routers without any storage at intermediate core nodes. The egress node, upon receiving the burst, disassembles the burst into packets and provides the packets to the upper layer. Basic architectures for core and edge routers in an OBS network have been studied elsewhere [28].

7.3 Threshold-Based Burst Assembly Technique

For burst assembly, we utilize a threshold 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.

The burst length affects the total number of contentions and the average number of packets lost per contention. For a higher threshold, the bursts will be longer, and there will be fewer bursts as well as fewer contentions. However in each contention, as each burst is longer, the average number of packets lost per contention will be higher. In the case of smaller bursts, there will be greater number of bursts in the network, and as a result, there will be a greater number of contentions; however fewer packets will be lost per contention. Thus, 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.

We compare the performance of different threshold schemes under the standard drop policy (DP) and the segmentation policy (SDP) for contention resolution. 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. In this chapter, we evaluate the following threshold-based QoS policies:

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.

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.

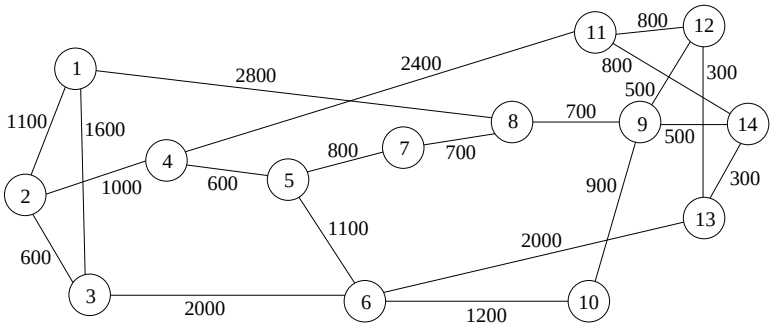


Figure 7.2. NSF Network with 14 nodes (distances in km).

7.4 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. 7.2 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.

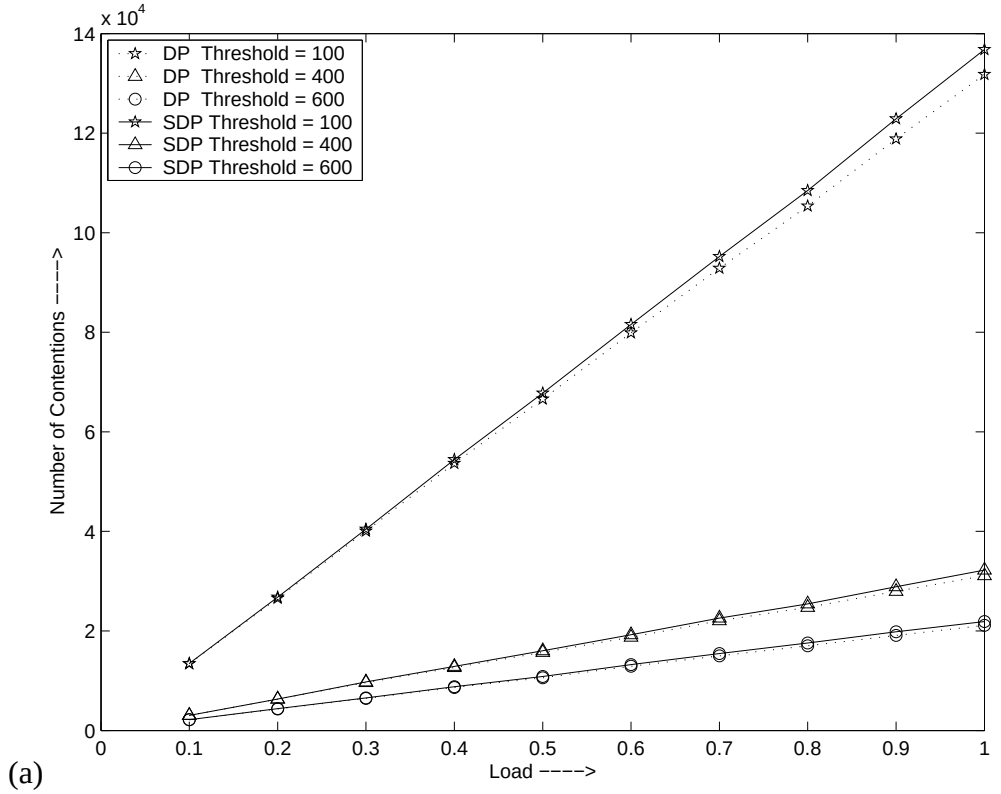
7.4.1 Single Threshold Without Burst Priority

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

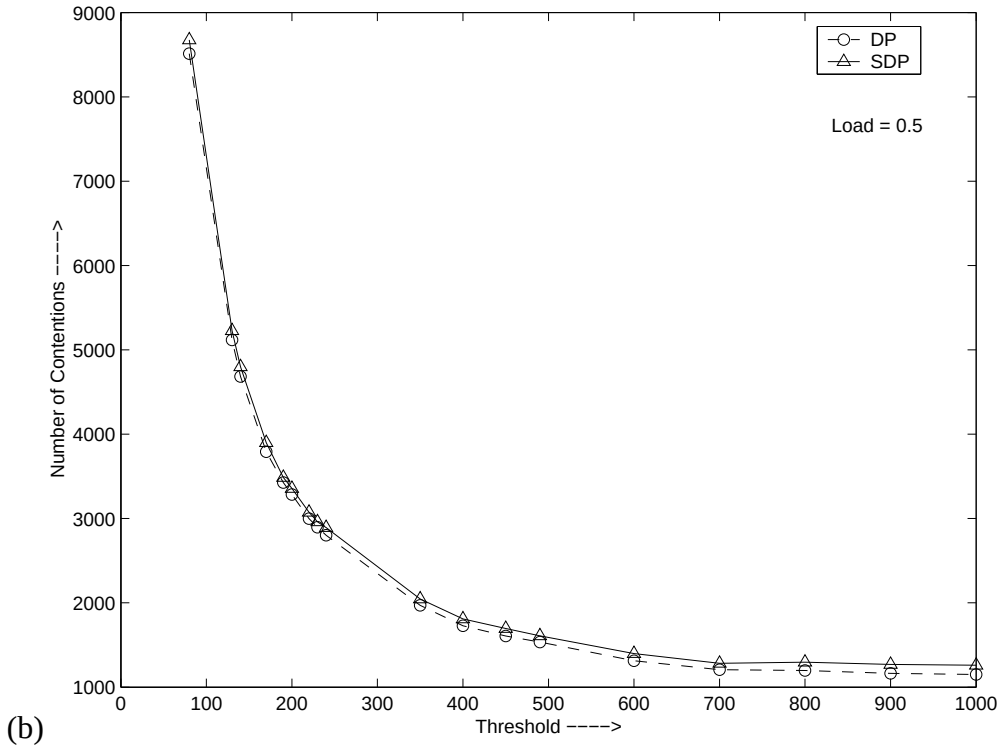
various loads and thresholds. From this single-threshold result we observe an optimum value of threshold for a given load and for a given network, for which the probability of packet loss will be minimum. Figures 7.3 and 7.4, give the performance of various parameters with DP or SDP as the contention resolution policy at the core.

Fig. 7.3(a) and Fig. 7.3(b) plot the load versus the total number of burst contentions. In this chapter, we simulate for 100 million (10^8) fixed-size packets. We observe that, as the load increases, the number of bursts in the network also increases, which leads to a higher number of contentions. In Fig. 7.3(a), we illustrate the total number of contentions for fixed threshold values of 100, 400, and 600 packets. We observe that the number of contentions increases with increases in load. Also, the number of contentions increases as the threshold value of the burst decreases. This result can be better understood by observing Fig. 7.3(b). We also observe that the number of contentions is slightly higher when SDP is employed as compared to when DP is used. The higher number of contentions is an effect of segmentation. For every contention between two bursts in DP, one of the bursts is dropped. In SDP, when the original burst is segmented, the contending burst continues forward in the network; hence the segmented burst may collide with another burst during its journey toward its destination, which in turn leads to a higher number of contentions in the networks.

Fig. 7.4(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. 7.4(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-



(a)



(b)

Figure 7.3. The graphs for DP and SDP with single threshold and no burst priority in the network. (a) Total number of burst contentions versus load. (b) Total number of burst contentions versus varying threshold values.

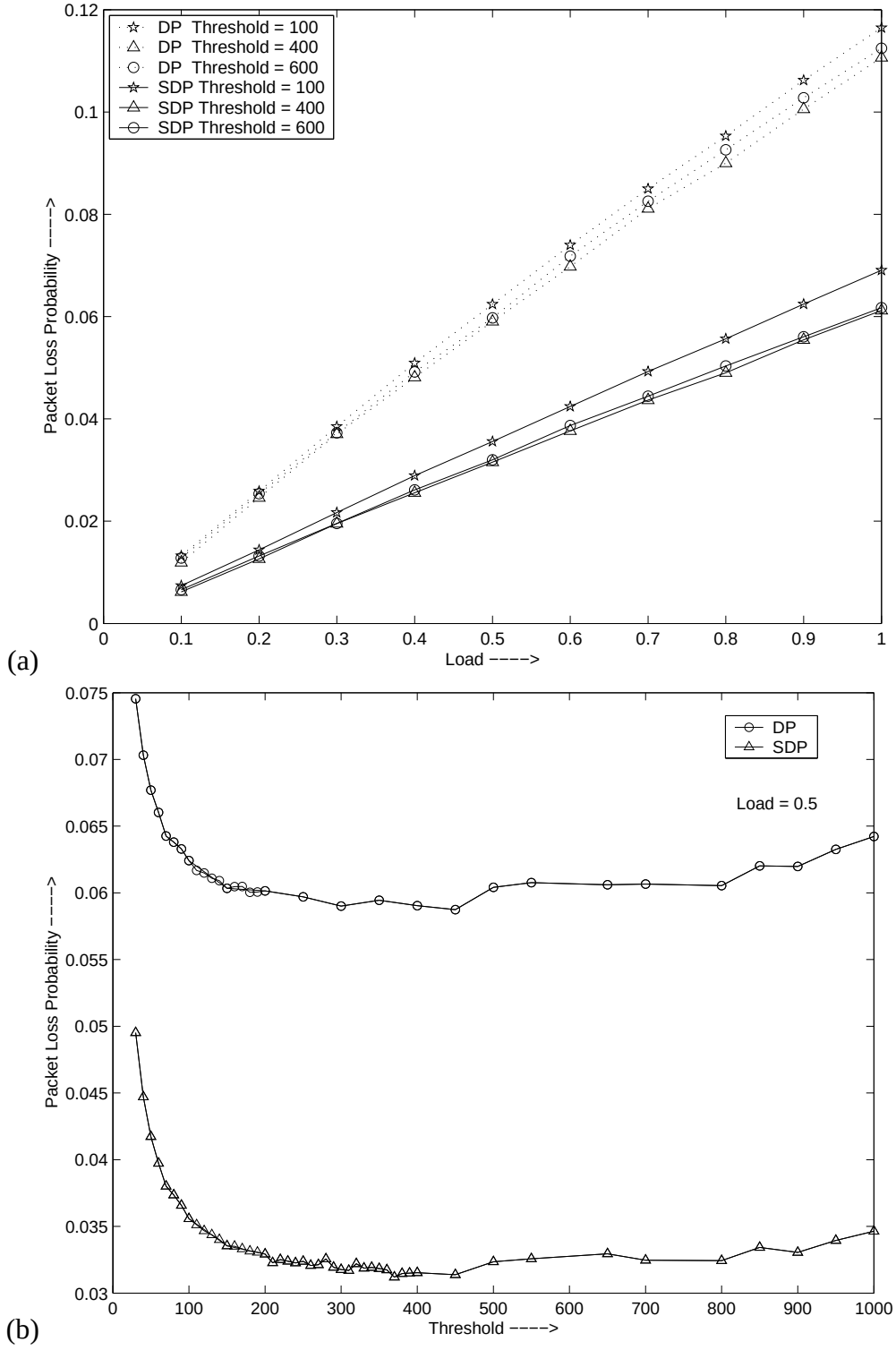


Figure 7.4. The graphs for DP and SDP with single threshold and no burst priority in the network. (a) Packet loss probability versus load. (b) Packet loss probability versus varying threshold values.

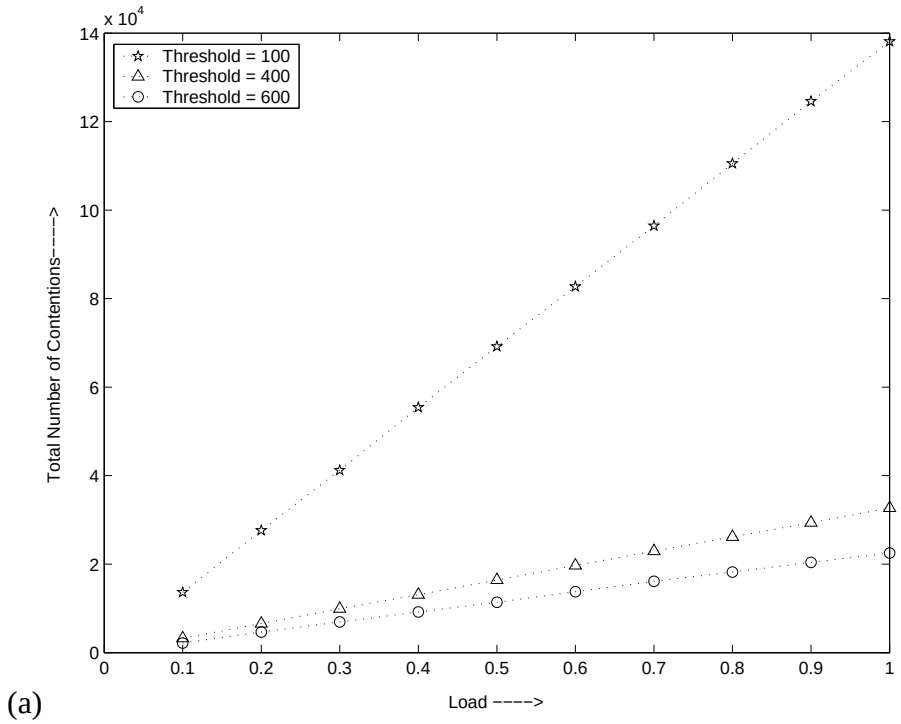
450 packets. After 450, the loss increases, since an increase in the threshold results in an increase in the average number of packets lost per contention. We choose 400 packets to be the optimal threshold value for the NSF network under a load range of 0 to 1 Erlang. The optimal threshold may vary based on the nodal degree of the network as well as the load range of the network.

7.4.2 Single Threshold With Burst Priority

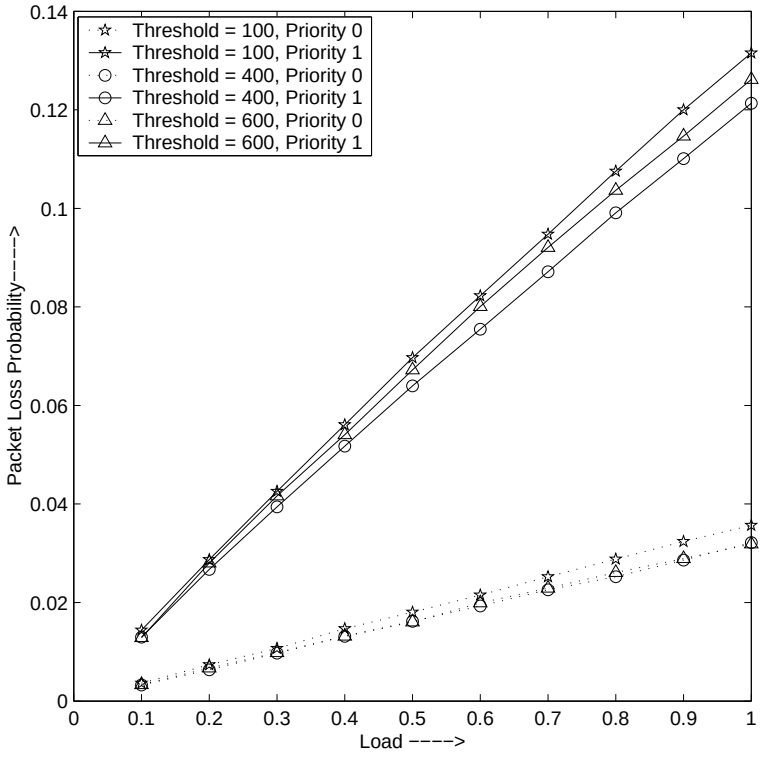
For the case of two burst priorities and a single threshold, we evaluate the packet loss probability and the number of contentions for variations in load and threshold. The two burst priorities are Priority 0 and Priority 1. Priority 0 represents higher-priority traffic. We use the optimum threshold value obtained from Fig. 7.4(b) as the threshold value, since it minimizes packet loss. Figs. 7.5(a)-(b) and 7.7, 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. 7.5(a) plots the total number of burst contentions versus load. We observe that, as the load increases, the total number of contentions increases. Also, as the threshold increases, the total number of contentions decreases, due to fewer bursts. Fig. 7.5(b) 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. 7.7. We observe that the loss of high-class packets is lower than that of low-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.



(a)



(b)

Figure 7.5. The graphs for SDP with single threshold and two burst priorities in the network. (a) Total number of burst contentions versus load. (b) Packet loss probability versus load for different threshold values.

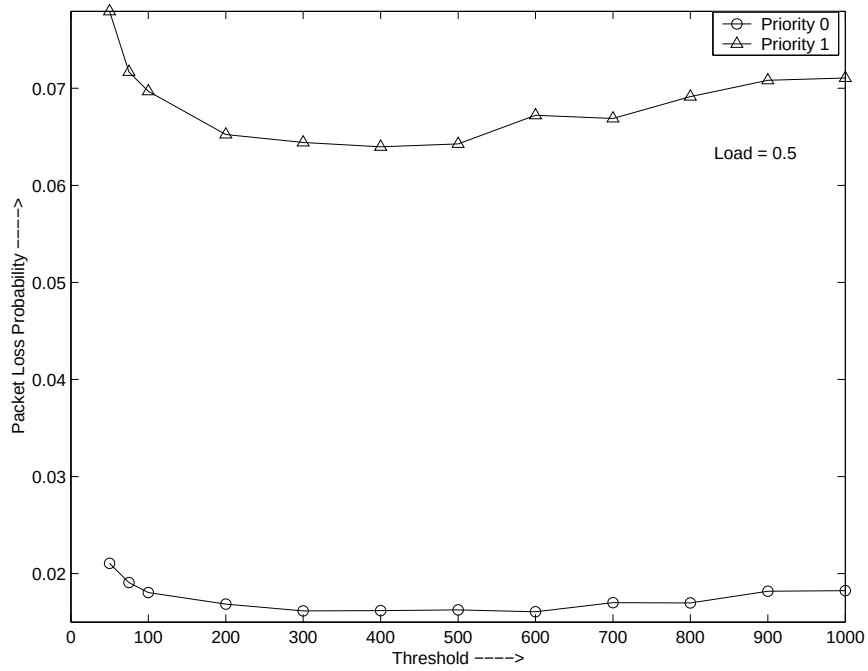


Figure 7.6. The graphs for SDP with single threshold and two burst priorities in the network. Packet loss probability versus threshold for both classes of packets at a load of 0.5 Erlang.

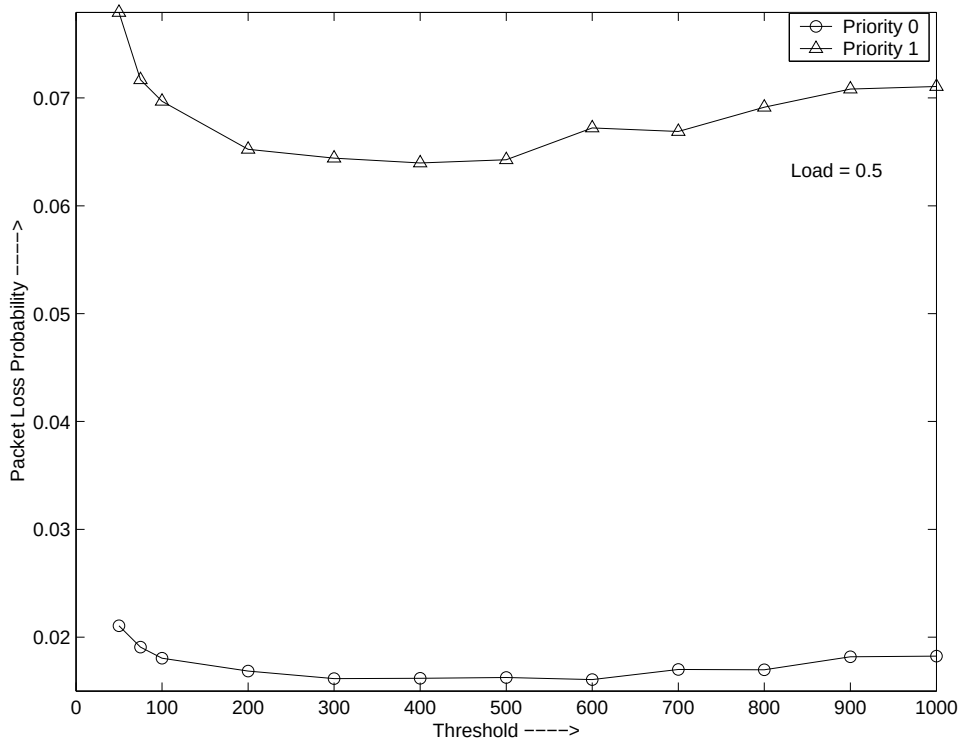


Figure 7.7. The graphs for SDP with single threshold and two burst priorities in the network. Packet loss probability versus threshold for both classes of packets at a load of 0.5 Erlang.

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

7.4.3 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 Figs. 7.8(a)-(b). 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.

Figure 7.8(a) plots the total number of burst contentions versus both threshold values. We observe that, as the threshold increases, the number of contentions decreases. In Fig. 7.8(b) we observe the packet loss probability for different values of threshold. Since there are no burst priorities in the network, during 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. 7.8(b) meet when both thresholds are equal. Since no priority is incorporated into the network, the loss is symmetrical for bursts of both threshold values.

7.4.4 Two Thresholds With Burst Priority:

Figures 7.9(a) and (b) show 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.

In Fig. 7.9(a) the total number of burst contentions is plotted versus both thresholds at a load of 0.5 Erlang. We observe that the number of contentions decrease as the threshold increases. Fig. 7.8(a) and Fig. 7.9(a) are similar with respect to the total number of burst contentions. Fig. 7.9(b) 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

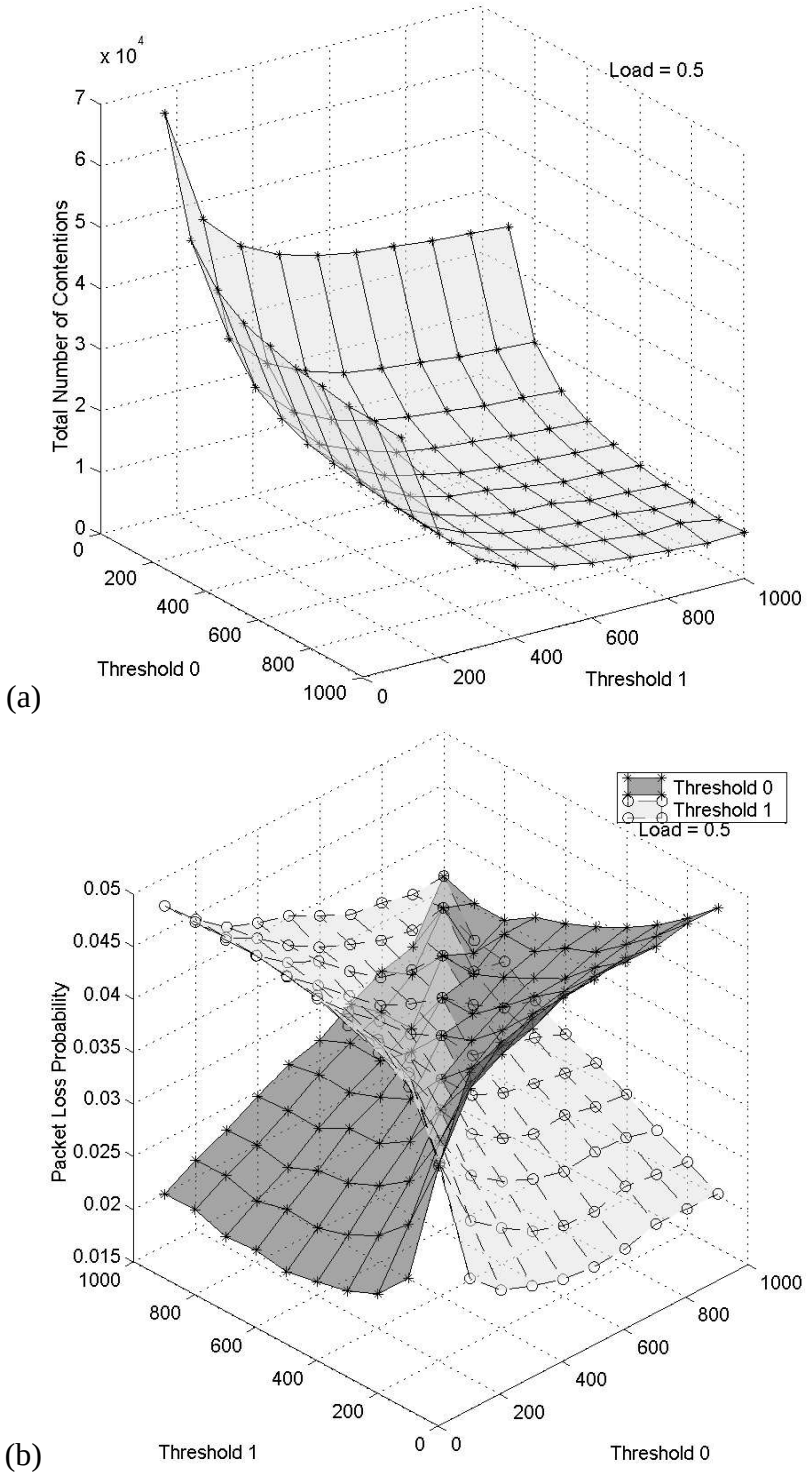
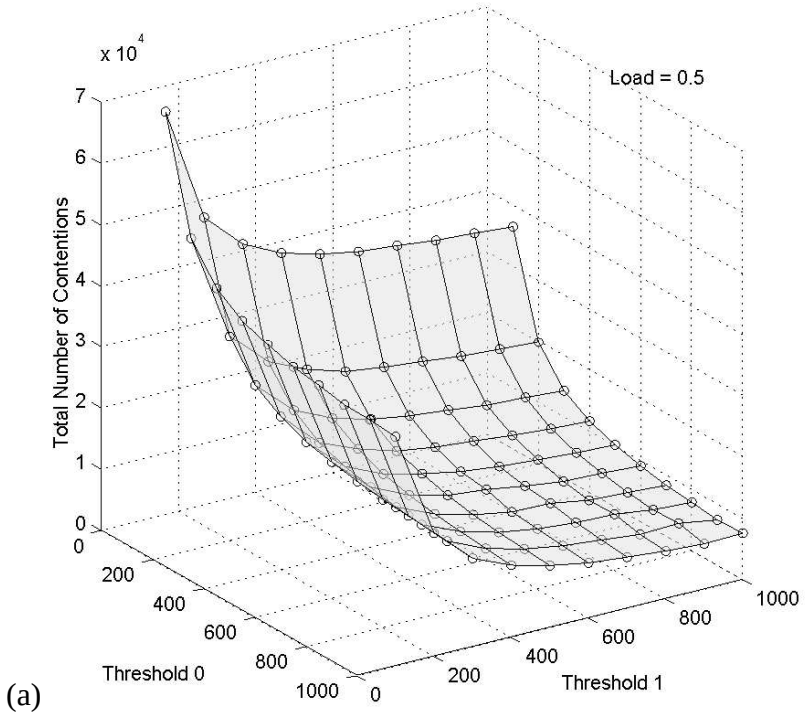
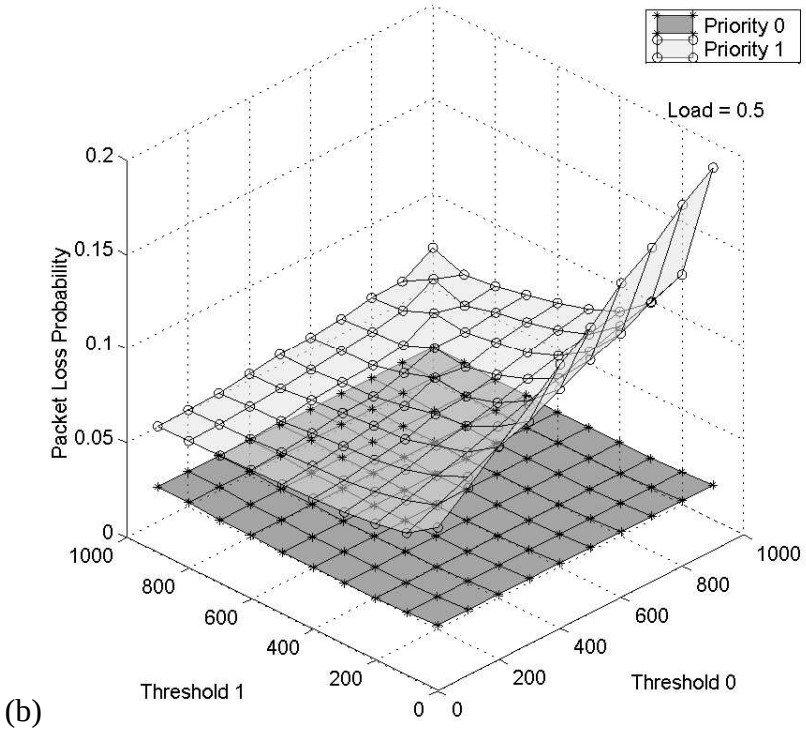


Figure 7.8. The graphs for SDP with two thresholds and no burst priority in the network (a) Total number of burst contentions versus varying both threshold values. (b) Packet loss probability versus varying both threshold values for both priorities.



(a)



(b)

Figure 7.9. The graphs for SDP with two threshold and two burst priorities in the network (a) Total number of burst contentions versus varying values for both thresholds. (b) Packet loss probability versus varying threshold values for both priorities.

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.

In general, we observe 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.

7.5 Conclusion

In this work, we considered an OBS network which uses the DR technique with burst segmentation. We investigated current timer-based and threshold-based burst assembly techniques, and we introduced a new threshold-based burst assembly technique to provide differentiated services for supporting QoS in the OBS network. We evaluated the relative performance of different threshold-based schemes for various threshold values and burst priorities, and we found that there is an optimal threshold value that minimizes the packet loss probability for a given network at a given load. We found that the optimal threshold range is between 380-430 packets for the NSF network under a load which ranges between 0 and 1 Erlang. By using fixed-size bursts of optimal threshold value, the packet loss can be minimized.

CHAPTER 8

INTERMEDIATE NODE INITIATED (INI) SIGNALING: A HYBRID RESERVATION TECHNIQUE FOR OPTICAL BURST-SWITCHED NETWORKS

8.1 Introduction

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. A detailed survey of the signaling techniques using a generalized framework is discussed in Section 2.5.

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 [151], in addition to containing the offset time. The offset time 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 Section 2.9). 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 chapter, we propose a hybrid signaling technique called *intermediate node initiated (INI)* signaling, and extend the proposed technique to provide differentiated signaling based on application requirements through *differentiated INI (DINI)* techniques. The DINI technique provides differentiation without introducing any additional offset time.

The remainder of the chapter is organized as follows. Section 8.2 describes extensions to the generalized OBS signaling framework proposed earlier in Section 2.6.1. Section 8.3 describes the proposed INI signaling technique. Section 8.4 extends the proposed INI signaling technique to provide differentiated signaling based on application requirements. In Section 8.6, we evaluate the end-to-end delay incurred by the different signaling techniques in OBS. Section 8.7 provides numerical results from simulation, and Section 8.8 concludes the chapter.

8.2 Extensions to the Generalized Signaling Framework

Signaling is a critical aspect that affects the performance of an network. For an OBS network, signaling is an important issue, since the core does not have any buffering mechanism. In this section, we extend the generalized signaling framework to include certain hybrid reservation schemes. By using the signaling framework, we can carefully evaluate various design param-

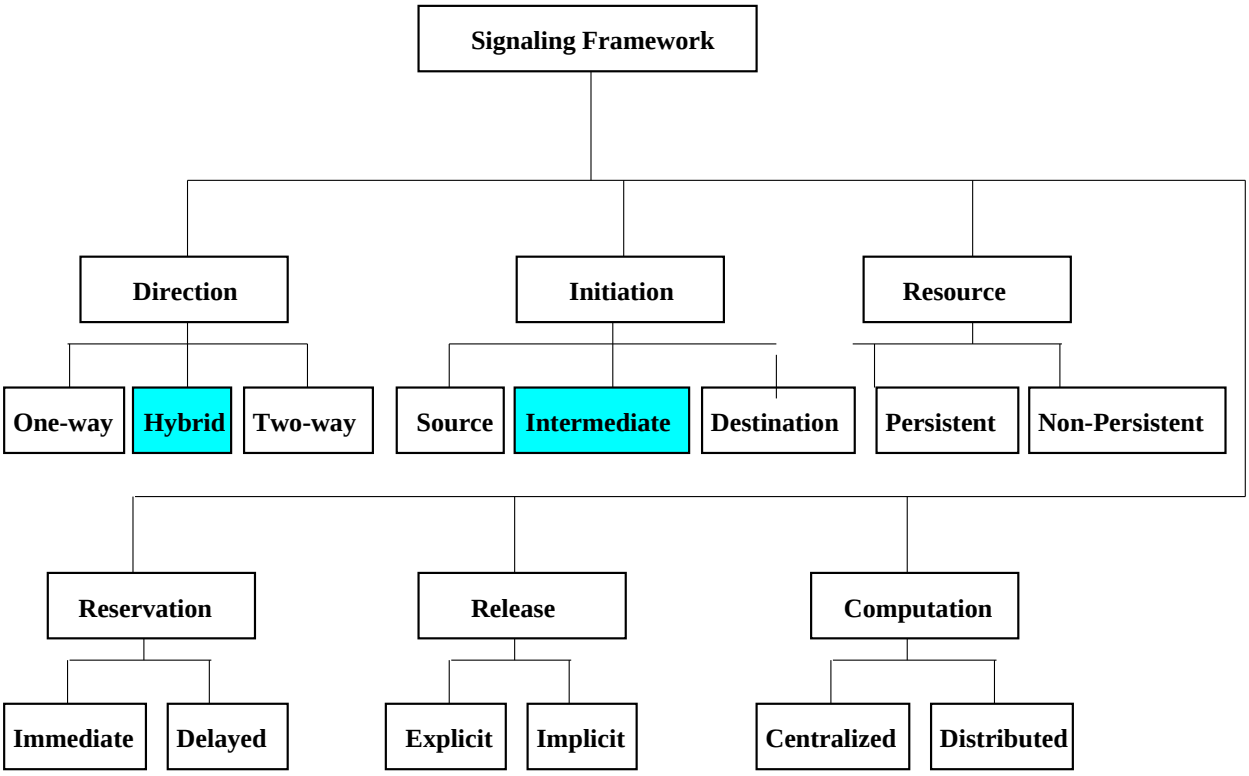


Figure 8.1. Generalized signaling framework.

eters before opting for a particular signaling technique, given the requirements of the data to be transmitted. Let us first briefly review the different parameters of the previously developed generalized signaling framework (Fig. 2.5). The following are the important parameters of the generalized signaling framework:

- *One-way or Two-way Connection Setup.*
- *Source-Initiated or Destination-Initiated Reservation.*
- *Persistent or Non-persistent Resource Request.*
- *Immediate Reservation or Delayed Reservation*
- *Explicit Release or Implicit Release*
- *Centralized or Distributed Signaling*

The details of each of the parameters are discussed in the Section 2.6.1. The following are the additions to the generalized signaling framework (Fig. 8.1) presented in Chapter 2.

- *One-way or Two-way or Hybrid (part two-way and part one-way)* In the hybrid signaling technique, the signaling is two-way based from the source to an initiating node (IN), and one-way based from the initiating node to the destination. We shall discuss this hybrid technique in detail in the next section. Based on the position of the initiating node, different loss and delay characteristics can be obtained. If the initiating node is closer to the source, performance is similar to one-way based techniques, such as JET, and if the initiating node is closer to the destination, performance is similar to two-way based techniques, such as TAW.

- *Source-Initiated or Destination-Initiated or Intermediate-Initiated Reservation:* In an *intermediate node initiated reservation*, typically the resources are reserved similar to destination-initiated reservation (DIR) from the source until the initiating node, and similar to source-initiated reservation (SIR) from the initiating node to the destination node.

In this chapter, we propose a new signaling technique, Intermediate Node Initiated (INI) signaling, which takes into account the advantages of both TAW and JET, and which provides the flexibility in meeting delay and loss tolerance requirements. The reservation request is initiated at an intermediate node, called the initiating node (IN). In the first part of the path, i.e., from source to the initiating node, the INI signaling technique works with an acknowledgment for the BHP similar to TAW. In the later part of the path, from the initiating node to destination, the INI signaling technique works without an acknowledgment, similar to JET.

8.3 Intermediate Node Initiated (INI) Signaling

To overcome the limitations of TAW and JET, we 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 8.2 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 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.

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

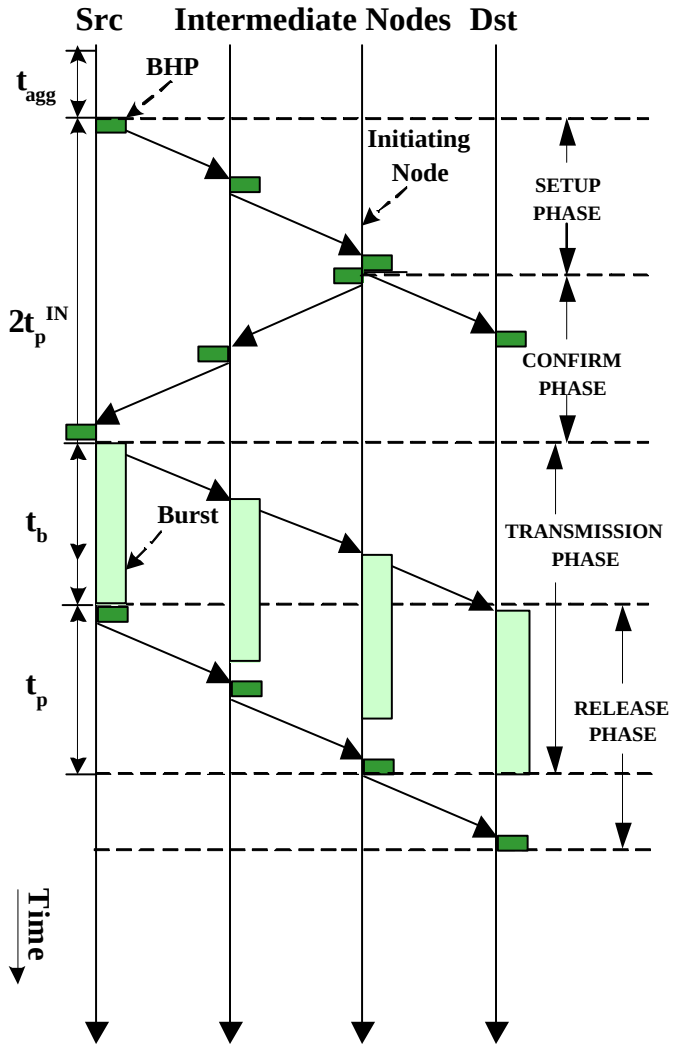


Figure 8.2. Intermediate Node Initiated (INI) Signaling Technique.

Table 8.1. 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	Intermediate	Exp./Imp.	Exp./Imp.	Flexible	Flexible

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 our simulations, we used delayed reservation.

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. 8.3, 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.

8.4 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.

8.5 Threshold-based Differentiated Intermediate Node Initiated (TDINI) Signaling

In the INI scheme ([152]), if we assume that the lengths of the burst are identical (fixed), the burst loss probability is same as the packet loss probability. If we consider a scenario in which the variance of the burst length (distributions) is very high, then the burst loss probability does not accurately represent the packet loss probability. For example, if 10% of the total bursts transmitted constitute 90% of the total number of packets being transmitted, it is important that these 10% of the bursts (relatively larger bursts) arrive at their destinations safely. That is, we would be better off losing the remaining 90% of bursts (relatively shorter bursts), which constitute only 10% of the total number of packets than losing relatively larger bursts. It is important to understand that low burst-loss probability does not necessarily translate into low packet-loss probability.

We introduce threshold-based differentiated intermediate-node initiated (TDINI) signaling [153], a variant of INI signaling in which the initiating node (IN) is determined based on the burst length. In the TDINI scheme, for every burst that is to be transmitted between a given source and a given destination, we choose one of the nodes on the path, from source to destination, to be the IN based on the following function:

$$f = \lfloor (l/l_{\max})(h+1) + 0.5 \rfloor$$

where l is the length of the burst to be transmitted, $l_{\max}$ is the maximum burst length and h is the number of hops from the source to the destination. The f^{th} node from the source is chosen to be the IN. In simpler words, the IN for lengthy bursts will be closer to the destination while the IN for shorter bursts will be closer to the source.

In the TDINI scheme (as in the case on INI), the BHP that is sent from the source collects the channel availability information at every node along its path until it reaches the IN. At the IN, a channel assignment algorithm is used to determine the time intervals for which each channel between the source and the IN needs to be reserved. The IN sends a "REPLY" packet to the source, which reserves the channels along the path from the IN to the

source for appropriate time intervals. If, in the case a channel is busy, a “FAIL” packet is sent to the IN asking it to release the already reserved resources. If the “REPLY” packet reaches the source successfully, then the source transmits the burst at the scheduled time. In the mean time, the BHP traveling from the IN to the destination reserves the channel along the path from the IN to the destination similar to JET scheme.

Ideally, we want larger bursts to reach the destination safely in order to reduce the packet-loss probability. Our idea of using burst-lengths to determine the IN that reduces the probability of dropping a lengthy burst, which in turn guarantees lower packet-loss probability. For longer bursts, the IN will be closer to the destination node, which means that a greater part of the path will be acknowledged, thereby guaranteeing a higher success rate. For smaller bursts, the IN will be closer to the source node, which means that greater part of the path will be un-acknowledged, thereby resulting a high burst-loss probability. But since smaller bursts constitute fewer packets compared to the total number of packets, there should not be any paramount concern about losing smaller bursts.

8.6 Analytical Delay Model

In this section, we develop 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.

$$T_{JET} = T_{JIT} = t_{agg} + t_{ot} + t_b + \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} \tag{8.1}$$

where,

$$t_{ot} = n t_{bhp} + t_{sw}. \tag{8.2}$$

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.

$$T_{TAG} = t_{agg} + nt_{fdl} + t_b + \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij}. \tag{8.3}$$

(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,

$$T_{TAW} = t_{agg} + 3 \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} + t_b + t_{ot}. \tag{8.4}$$

Also,

$$t_{ot} = 0 \; if \;\; 2 \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} \geq (n+2) t_{bhp} + t_{sw}. \tag{8.5}$$

(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.

$$T_{INI} = t_{agg} + 2 \sum_{l^{ij} \in R_{sk}}^{n} t_p^{ij} + \sum_{l^{ij} \in R_{sd}}^{n} t_p^{ij} + t_b + t_{ot} \tag{8.6}$$

where,

$$t_{ot} = 0 \; if \;\; 2 \sum_{l^{ij} \in R_{ks}}^{n} t_p^{ij} \geq \; mt_{bhp} + t_{sw} \tag{8.7}$$

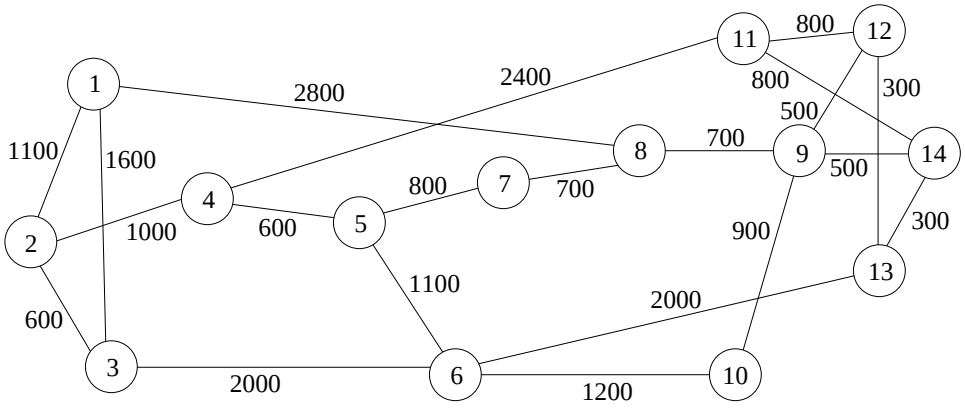


Figure 8.3. 14-node NSF USA backbone network topology (distance in km).

else,

$$t_{ot} = (mt_{bhp} + t_{sw}) - 2 \sum_{l^{ij} \in R_{k_s}}^{n} t_p^{ij}. \tag{8.8}$$

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

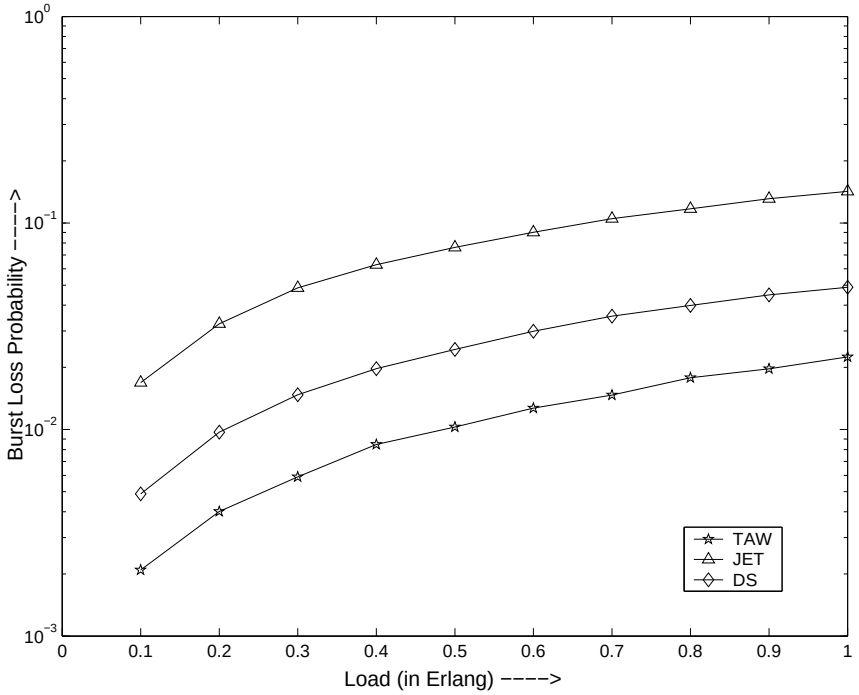
Hence,

$$T_{JET} \leq T_{INI} \leq T_{TAW}. \tag{8.9}$$

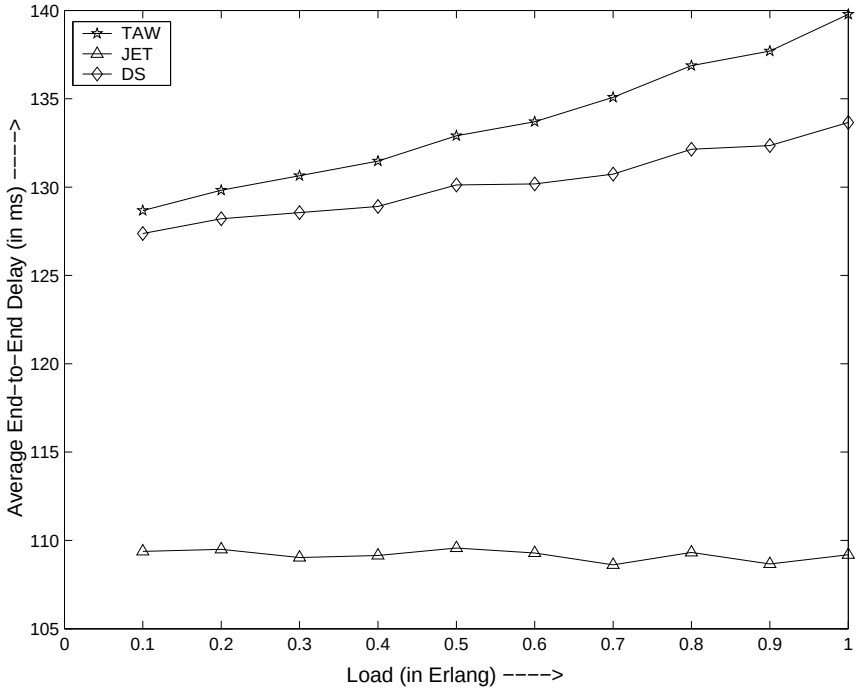
8.7 Numerical Results

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. Figure 8.3 shows the 14-node NSFNET on which the simulation is implemented.

Figures 8.5(a) and 8.5(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. 8.5 (a) and

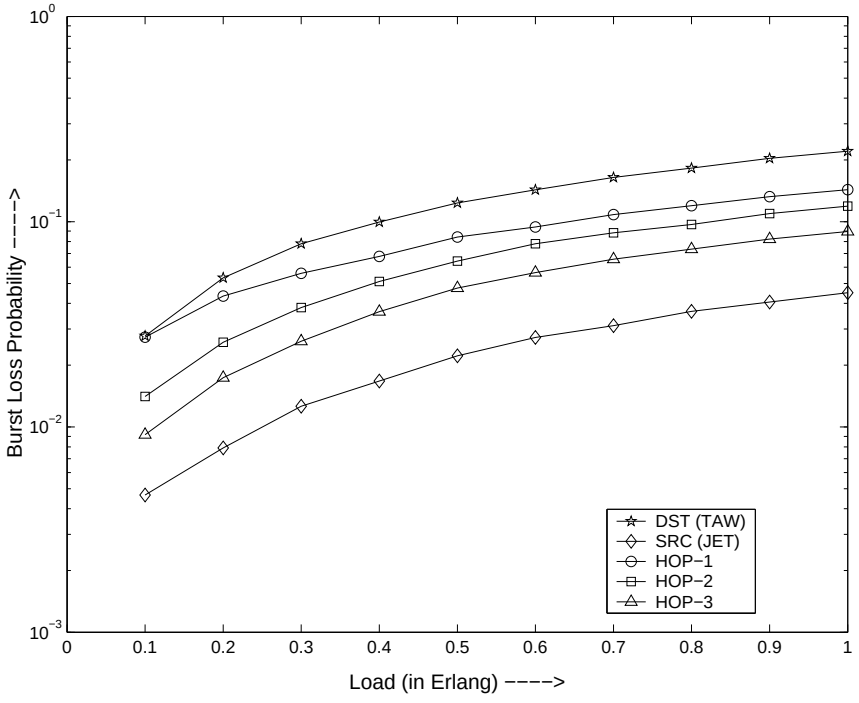


(a)

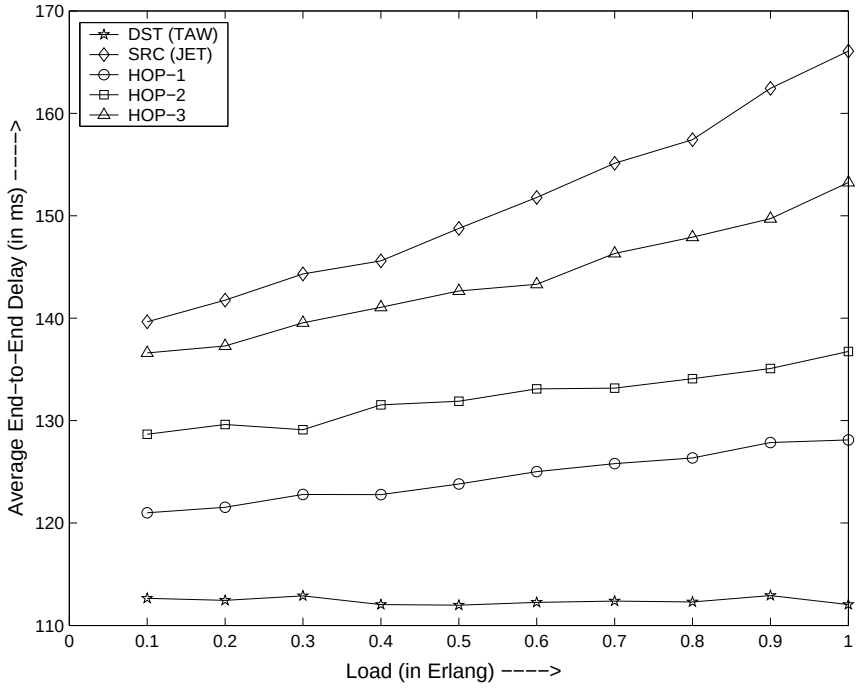


(b)

Figure 8.4. (a) Burst loss probability versus load, and (b) Average end-to-end delay versus load, for JET, TAW, and INI with the initiating node is at the center hop.



(a)



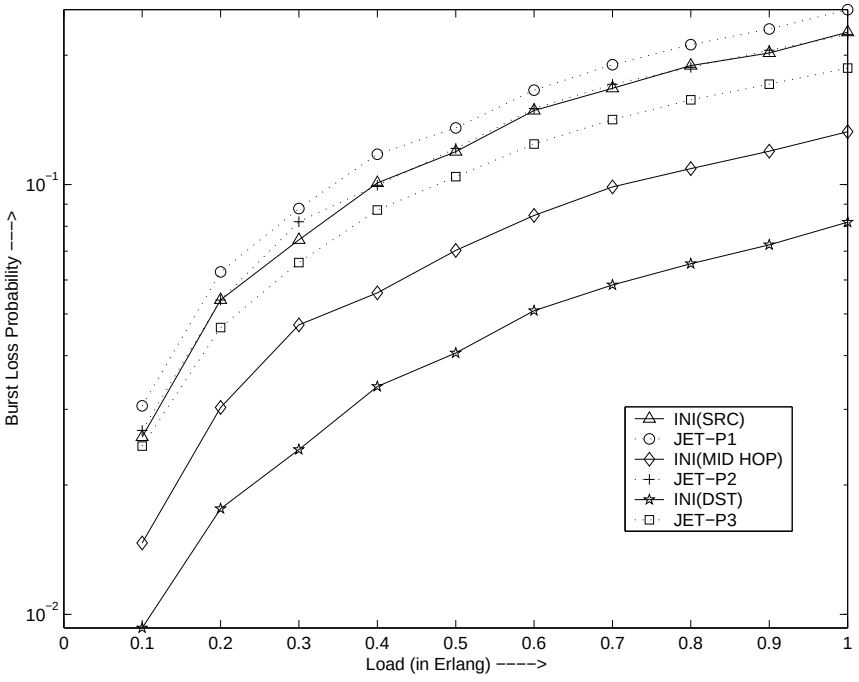
(b)

Figure 8.5. (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.

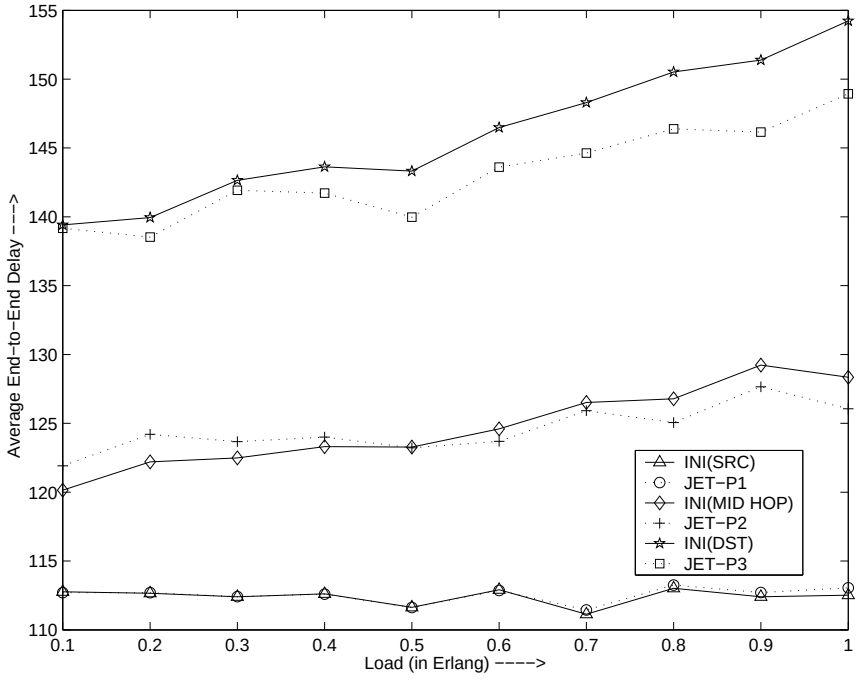
8.5 (b), only paths that are more than or equal to three hop counts are considered to show the effect of INI signaling technique. We observe that the loss probability decreases as the initiating node moves away from the source. If the initiating node is chosen closer to the source, a greater part of 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.

Figures 8.6(a) and 8.6(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 [95]. 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 the network until resources have been reserved between the source node and the initiating node.

Figures 8.7(a) and (b) depict load versus packet loss probability and average end-to-end delay, respectively using TDINI. We classify the bursts into three groups based on their lengths, namely, large-size bursts (top 1/3), medium-size burst (mid 1/3), and small-sized bursts (lower 1/3). We observe that the packet-loss probability of longer bursts is much lower than that of the shorter bursts. As a trade-off, the longer burst lengths have higher average end-to-end delay, since a larger portion of their path is two-way acknowledged and vice versa. Medium-sized bursts enjoy average loss and delay performances.

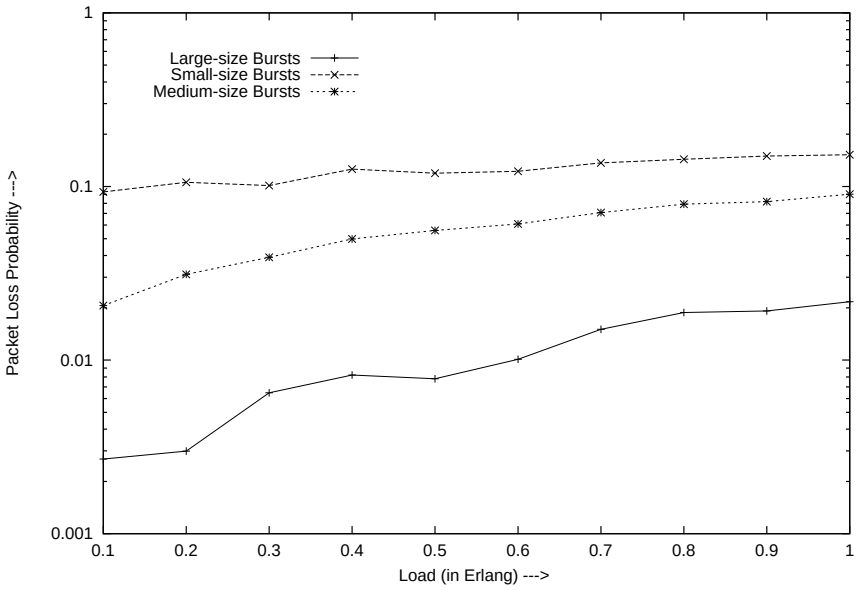


(a)

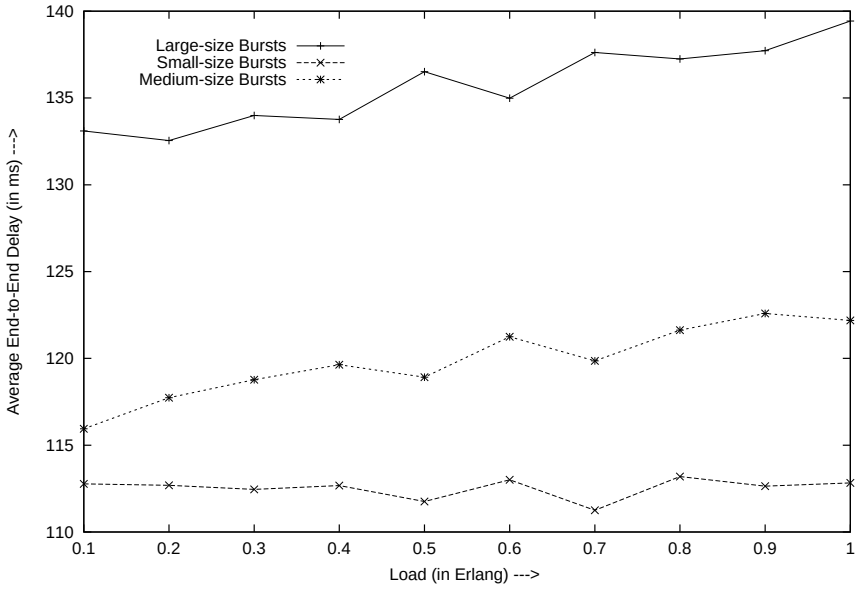


(b)

Figure 8.6. (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.



(a)



(b)

Figure 8.7. (a) Packet loss probability versus load and (b) Average end-to-end delay versus load.

8.8 Conclusion

In this chapter, we introduced the intermediate node initiated signaling technique for an OBS network. 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 proposed hybrid technique is a flexible solution suitable for handling the varying traffic demands of the next-generation optical network.

TAW and JET are discussed in detail, and the advantages and disadvantages of each technique are discussed. We introduced a new signaling technique for optical burst switching, intermediate node initiated signaling. We described the working principle of the INI signaling technique, and its advantages over existing techniques such as TAW and JET. Through simulation, we showed that the INI signaling technique performs better than TAW and JET and the generalized INI framework includes both TAW and JET. We also showed how the Differentiated INI signaling technique can be used to provide QoS and verified the technique through simulations. For comparable values of offset-time, we found that DINI out-performs the traditional offset-based QoS scheme. Another extension of INI, referred to as Threshold-based DINI has also been proposed to differentiate traffic (bursts) based on their burst lengths.

CHAPTER 9
CONCLUSION

The amount of traffic being transmitted over existing networks is rising at an unprecedented rate, and telecommunications and data communications companies are racing to provide the means for meeting these demands. Optical burst switches are the engines for high-speed Internet transport on optical networks. Optical burst switching combines the advantages of packet switching and circuit switching in a single network. Data and control information are sent through different wavelength channels in a WDM system. When bursts and headers are sent separately on different channels, new protocols are necessary to avoid burst loss. In this dissertation, we presented several architectures and protocols for solving some of the fundamental challenges faced by optical burst-switched networks. In this chapter, we will summarize the contributions of this work, and we will provide some directions for future research.

9.1 Summary of Contributions

In Chapter 2, we provided a survey of the current literature on the fundamental issues in optical burst switching, such as network architecture, burst assembly, routing and wavelength assignment, edge scheduling, signaling, channel scheduling, contention resolution, and quality of service.

In Chapter 3, we introduced the concept of burst segmentation for contention resolution in optical burst switched networks, and we investigated a number of different policies with and without segmentation and deflection. The segmentation policies perform better than the standard dropping policy, and offer the best performance at high loads. The policies which

incorporate deflection tend to perform better at low loads, while deflection is not as effective at high loads.

In Chapter 4, we considered burst segmentation and FDLs with wavelength conversion for burst scheduling in optical burst-switched networks, and we proposed 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, while the 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.

In Chapter 5, we introduced the concept of prioritized contention resolution through prioritized burst segmentation and deflection to provide QoS in the optical burst-switched core network. The prioritized contention resolution policies can provide QoS with 100% class isolation without requiring any additional offset times. An analytical model for prioritized burst segmentation was developed to calculate the packet loss probabilities for a two-priority network, and the model was verified through simulation. The high-priority bursts have significantly lower losses and delay then the low-priority bursts, and the schemes which incorporate deflection tend to perform better than the schemes with limited deflection or no deflection. Also, prioritized burst segmentation is easily scalable in order to support multiple priorities in an all-optical burst-switched network.

In Chapter 6, we introduced the concept of composite burst assembly to handle the differentiated service requirements of the IP packets at edge nodes of the optical burst-switched network, and we described a generalized framework for burst assembly.We considered four different burst assembly approaches and evaluated their performance in terms of delay and

loss. We observe that approaches with composite bursts perform better than approaches with single-class bursts with respect to providing differentiated QoS for different classes of packets. This was verified by the analytical model results. The developed model can be useful for selecting the class ratios for composite bursts in a manner which can satisfy the packet loss requirement. In order to further reduce the packet loss, the proposed techniques can be employed in conjunction with all-optical wavelength conversion and buffering through fiber delay lines.

In Chapter 7, we considered an OBS network which uses the DR technique with burst segmentation. We investigated current timer-based and threshold-based burst assembly techniques, and we introduced a new threshold-based burst assembly technique to provide differentiated services for supporting QoS in the OBS network. We evaluated the relative performance of different threshold-based schemes for various threshold values and burst priorities, and we found that there is an optimal threshold value that minimizes the packet loss probability for a given network at a given load. We found that the optimal threshold range is between 380-430 packets for the NSF network under a load which ranges between 0 and 1 Erlang. By using fixed-size bursts of optimal threshold value, the packet loss can be minimized.

In Chapter 8, we introduced the intermediate node initiated signaling technique for an OBS network. 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 proposed hybrid technique is a flexible solution suitable for handling the varying traffic demands of the next generation optical network. We described the working principle of the INI signaling technique, and its advantages over existing techniques like TAW and JET. Through simulation, we showed that the INI signaling technique performs better than TAW and JET. We also showed how the Differentiated INI signaling technique can be used to provide QoS and verified the technique through simulations. For comparable values of offset-time, we found that DINI out-performs the traditional offset-based QoS scheme.

9.2 Future Work

The following are some of the possible areas of future work based on individual chapters in the dissertation.

In Chapter 3, we have considered only one alternate output port for deflection, an area for future work is the investigation of policies which consider multiple alternate output ports and in which the selection criteria is based on load and shortest path may also be considered.

An interesting area of future work for channel scheduling (Chapter 4), would be to implement the preemptive scheduling algorithms for providing QoS support in the optical burst-switched networks. Also, to effectively evaluate the quality of service offered by various priority policies, a retransmission scheme for dropped packets could be implemented in order to measure end-to-end delay. A reasonable approach would be to implement a TCP layer on top of the optical burst-switched layer. In such an implementation, it would also be useful to evaluate how TCP layer congestion control schemes react to and interact with various contention resolution schemes [154, 155, 156, 157, 158].

In Chapters 5 and 6, in order to further reduce the packet loss, the proposed techniques can be employed in conjunction with all-optical wavelength conversion, and buffering through fiber delay lines. Also, extending the proposed assembly framework to handle variable-sized packets is important. Composite burst assembly is also very important when the core network supports both multicast and unicast application. In this situation, the edge has to differentiate the arriving packets based on their destination egress nodes, their quality of service class, and also based on whether they are multicast or unicast applications.

Possible areas of future work in Chapter 7 are to analyze the end-to-end delay for the threshold-based schemes, to evaluate the performance in the case of more than two packet classes, and to investigate timer-based assembly techniques to support delay-based QoS. By combining both timer-based and threshold-based scheme, it may be possible to provide minimal loss while also guaranteeing end-to-end delay.

In Chapter 8, an area of future work is to study the performance of the INI signaling technique with wavelength conversion in a multi-wavelength system and combine with deflection of BHPs during signaling to improve channel utilization and monitor the delay trade-off. Also, the performance of INI can be improved by implementing void filling, i.e., utilizing channel gaps between existing reservations, for reservation. Also, to develop accurate analytical loss and delay model for both INI and DINI.

9.3 OBS: Candidate for Supporting the Next-Generation Optical Internet

As the Internet continues to experiencing enormous growth, it will be expected to support a growing number of applications, such as Internet telephony, video conferencing, and video distribution. To support these different applications, the underlying network must be capable of not only providing the required bandwidth, but also providing guarantees with respect to quality of service, security, and fault tolerance. A WDM network can provide the necessary bandwidth to support a wide range of multimedia Internet services; however, further work is required towards the development of new switch and router architectures, along with appropriate protocols, to enable service guarantees.

Optical burst switching (OBS) is one of the most promising transport technologies for supporting the next-generation optical Internet. Since OBS has the versatility of supporting long duration connections request (similar dynamic lightpaths requests) using two-way based signaling techniques, such as TAW and INI (destination as initiating node), and at the same time OBS can support short duration requests (similar to packets) using one-way based signaling techniques, such as TAG, JET, JIT, and INI (source as initiating node). There is a need for continued research so as to understand several fundamental issues, such as TCP over OBS, fault tolerance in OBS, and security in OBS networks. In order to keep pace with the development of new applications, such as grid computing and peer-to-peer computing, we must develop and take advantage of new technologies so that we can ensure that this tremendous growth will continue well into the future.

REFERENCES

[1] S. Chatterjee and S. Pawlowski, "All-optical networks," *Communications of the ACM*, vol. 42, no. 6, pp. 74–83, June 1999.

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

[3] R.C. Alferness, H. Kogelnik, and T.H. Wood, "The evolution of optical systems: Optics everywhere," *Bell Labs Technical Journal*, vol. 5, no. 1, Jan-March 2000.

[4] B.Mukherjee, "WDM optical communication networks: Progress and challenges," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, OCTOBER 2000.

[5] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WANs," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171–1182, 1992.

[6] 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*, vol. 1, no. 1, January 2000.

[7] S. Yao, B. Mukherjee, and S. Dixit, "Advances in photonic packet switching: An overview," *IEEE Communications Magazine*, vol. 38, no. 2, pp. 84–94, February 2000.

[8] S. Yao, S. J. B. Yoo, B. Mukherjee, and S. Dixit, "All-optical packet switching for metropolitan area networks: Opportunities and challenges," in *IEEE Communications Magazine*, March 2001, vol. 39, pp. 142–148.

[9] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, "A unified study of contention-resolution schemes in optical packet-switched networks," in *IEEE/OSA Journal of Lightwave Technology*, March 2003.

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

[11] M. J. O'Mahony, D. Simeonidou, D. K. Hunter, and A. Tzanakaki, "The application of optical packet switching in future communication networks," *IEEE Communications Magazine*, vol. 39, pp. 128–135, March 2001.

[12] D. K. Hunter and I. Andronovic, "Approaches to optical Internet packet switching," *IEEE Communications Magazine*, vol. 38, pp. 116–122, September 2000.

[13] F. Callegati, A. C. Cankaya, Y. Xiong, and M. Vandenhoute, "Design issues of optical IP routers for Internet backbone applications," *IEEE Communications Magazine*, vol. 37, pp. 124–128, December 1999.

[14] A. Jourdan, D. Chiaroni, E. Dotaro, G. J. Eilenberger, F. Masetti, and M. Renaud, "The perspective of optical packet switching in IP dominant backbone and metropolitan networks," *IEEE Communications Magazine*, vol. 39, pp. 136–141, March 2001.

[15] E. Haselton, "A PCM frame switching concept leading to burst switching network architecture," *IEEE Communications Magazine*, vol. 21, pp. 13–19, June 1983.

[16] S. Amstutz, "Burst switching - an introduction," *IEEE Communications Magazine*, vol. 21, pp. 36–42, November 1983.

[17] S. Amstutz, "Burst switching - an update," *IEEE Communications Magazine*, vol. 21, pp. 50–57, September 1989.

[18] D. K. Hunter et al., "WASPNET: A wavelength switched packet network," *IEEE Communications Magazine*, pp. 120–29, March 1999.

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

[20] L. Xu, H.G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 136–142, January 2001.

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

[22] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," *IEEE Communications Magazine*, vol. 38, no. 9, pp. 104–114, September 2000.

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

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

[25] G. G. Xie and S. S. Lam, "Real-time block transfer under a link sharing hierarchy," *IEEE/ACM Transactions on Networking*, 1996.

[26] 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*, vol. 5, pp. 705–718, October 1997.

[27] I. Widjaja, "Performance analysis of burst admission control protocols," *IEEE Proc. Commun.*, vol. 142, pp. 7–14, February 1995.

[28] Y. Xiong, M. Vanderhoute, and H.C. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838–1854, October 2000.

[29] 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.

[30] S. Verma, H. Chaskar, and R. Ravikanth, "Optical burst switching: a viable solution for terabit IP backbone," *IEEE Network*, vol. 14, no. 6, pp. 48–53, November 2000.

[31] 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.

[32] M. Dueser and P. Bayvel, "Analysis of a dynamically wavelength-routed optical burst switched network architecture," *IEEE/OSA Journal of Lightwave Technology*, vol. 20, no. 4, pp. 574–586, April 2002.

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

[34] M. Duser and P. Bayvel, "Performance of a dynamically wavelength-routed optical burst switched network," in *Proceedings, IEEE Globecom*, November 2001, vol. 4, pp. 2139–2143.

[35] 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*, vol. 21, no. 7, pp. 1198–1209, September 2003.

[36] X. Cao, J. Li, Y. Chen, and C. Qiao, "TCP/IP packets assembly over optical burst switching network," in *Proceedings, IEEE Globecom*, November 2002, vol. 3, pp. 2808–2812.

[37] 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*, November 1999, pp. 339–352.

[38] 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*, vol. 40, no. 2, pp. 82–89, February 2002.

[39] 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)*, 2001, pp. 138–143.

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

[41] X. Yu, Y. Chen, and C. Qiao, "Performance evaluation of optical burst switching with assembled burst traffic input," in *Proceedings, IEEE Globecom*, November 2002, vol. 3, pp. 2318–2322.

[42] M. Izal and J. Aracil, "On the influence of self-similarity on optical burst switching traffic," in *Proceedings, IEEE Globecom*, November 2002, vol. 3, pp. 2308–2312.

[43] A. Ge, F Callegati, and L. Tamil, "On optical burst switching and self-similar traffic," *IEEE Communications Letters*, vol. 4, no. 3, March 2000.

[44] Jun Kyun Choi and et. al, "Extension of gsmp for optical burst switching," *IETF draft-choi-gsmp-optical-extension-00.txt*, 2000.

[45] O. Gerstel and R. Ramaswami, "Optical layer surivivability: An implementation perspective," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1885–1899, October 2000.

[46] S. Ramamurthy and B. Mukherjee, "Survivable WDM networks. Part I - Protection," in *Proceedings, IEEE Infocom*, 1999, vol. 2, pp. 744–751.

[47] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 21, no. 4, pp. 870–883, April 2003.

[48] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: A new approach to the design of WDM-based networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 800–809, May 2002.

[49] F. Ricciato, S. Salsano, and M. Listanti, "Optimal routing for protection and restoration in an optical network," *Photonic Networks and Communications*, vol. 4, no. 3-4, pp. 409–422, July-December 2002.

[50] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, no. 6, pp. 16–23, November-December 2000.

[51] P.-H. Ho and H. T. Mouftah, "A framework for service guaranteed shared protection in WDM mesh networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 97–103, February 2002.

[52] D. Xu, Y. Xiong, and C. Qiao, "Novel algorithms for shared segment protection," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1320–1331, October 2003.

[53] Y. Xiong, D. Xu, and C. Qiao, "Achieving fast and bandwidth-efficient shared-path protection," *IEEE/OSA Journal of Lightwave Technology*, vol. 21, no. 2, pp. 365–371, October 2003.

[54] D. Griffith and S. Lee, "A 1 + 1 protection architecture for optical burst switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 9, 2003.

[55] S. Oh, Y. Kim, , and M. Yoo, "Path restoration schemes in the optical burst switched Internet," in *Proceedings, COIN-PS*, July 2002.

[56] H.S. Lim and H.S. Park, "A rapid recovery scheme considering optical burst switching property," in *Proceedings, COIN-PS*, July 2002.

[57] 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.

[58] Y. Wang and Z. Wang, "Explicit routing algorithms for Internet traffic engineering," in *Proceedings, IEEE ICCCN*, September 1999.

[59] G.P.V. Thodime, V. M. Vokkarane, and J. P. Jue, "Dynamic congestion-based load balanced routing in optical burst-switched networks," in *Proceedings, IEEE Globecom*, December 2003, vol. 5, pp. 2694–2698.

[60] J. Li, G. Mohan, and K. C. Chua, "Load balancing using adaptive alternate routing in IP-over-WDM optical burst switching networks," in *Proceedings, SPIE OptiComm*, October 2003.

[61] B. Chen and J. Wang, "Hybrid switching and p-routing for optical burst switching networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1071–1080, September 2003.

[62] J. A. White, R. S. Tucker, and K. Long, "Merit-based scheduling algorithm for optical burst switching," in *Proceedings, COIN-PS*, July 2002.

[63] B. C. Kim, J. H. Lee, Y. Z. Cho, and D. Montgomery, "Performance of optical burst switching techniques in multi-hop networks," in *Proceedings, IEEE Globecom*, November 2002.

[64] S.K. Tan, G. Mohan, and K.C. Chua, "Link scheduling state information based offset management in WDM optical burst switching networks," *Elsevier Computer Networks*, 2004.

[65] I. Ogushi, S. Arakawa, M. Murata, and K. Kitayama, "Parallel reservation protocols for achieving fairness in optical burst switching," in *Proceedings, IEEE Workshop on High Performance Switching and Routing*, November 2000.

[66] Y. Chen, M. Hamdi, and D.H.K. Tsang, "Proportional QoS over OBS network," in *Proceedings, IEEE Globecom*, November 2001, vol. 3, pp. 1510–1514.

[67] P. Ferguson and G. Huston, *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, John Wiley & Sons, Inc., 1998.

[68] Tachibana, T. Ajima, and S. Kasahara, "Round-robin burst assembly and constant transmission scheduling for optical burst switching networks," in *Technical Report No. 2003003, Graduate School of Information Science, NAIST, Japan*, April 2003.

[69] J. Li and C. Qiao, "Schedule burst proactively for optical burst switching networks," in *Proceedings, IEEE Globecom*, December 2003, pp. 2787–2791.

[70] 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.

[71] 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*, vol. 1, no. 3, pp. 207–218, 1999.

[72] 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*, vol. 42, no. 2, pp. 568–585, February 2003.

[73] 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)*, February 1999, vol. 3, pp. 180–182.

[74] 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*, 2002, vol. 3, pp. 2736–2740.

[75] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient channel scheduling algorithms in optical burst switched networks," in *Proceedings, IEEE Infocom*, March 2003, vol. 3, pp. 2268–2278.

[76] 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.

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

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

[79] V. M. Vokkarane and J. P. Jue, "Burst segmentation: An approach for reducing packet loss in optical burst switched networks," *SPIE Optical Networks Magazine*, vol. 4, no. 6, pp. 81–89, November-December 2003.

[80] 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)*, May 2003, vol. 2, pp. 1443–1447.

[81] V. M. Vokkarane and J. P. Jue, "Segmentation-based non-preemptive scheduling algorithms for optical burst-switched networks," in *Proceedings, First International Work-*

shop on Optical Burst Switching (WOBS), co-located with OptiComm 2003, October 2003.

[82] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 12, pp. 2081–2094, December 1998.

[83] 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*, vol. 16, no. 10, pp. 1725–1736, October 1998.

[84] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and et al., "CORD: Contention resolution by delay lines," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 1014–1029, June 1996.

[85] Z. Haas, "The 'Staggering Switch': An electronically controlled optical packet switch," *IEEE/OSA Journal of Lightwave Technology*, vol. 11, no. 5/6, pp. 925–936, May/June 1993.

[86] I. Chlamtac, A. Fumagalli, and C.-J. Suh, "Multibuffer delay line architectures for efficient contention resolution in optical switching nodes," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2089–2098, December 2000.

[87] 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*, vol. 16, no. 10, pp. 1725–1736, October 1998.

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

[89] 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)*, 1996, pp. 106–107.

[90] 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*, vol. 16, no. 10, pp. 1737–1745, October 1998.

[91] 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*, vol. 19, no. 9, pp. 1241–1254, September 2001.

[92] 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.

[93] F. Callegati, "Optical buffers for variable length packets," *IEEE Communications Letters*, vol. 4, no. 9, pp. 292–294, September 2000.

[94] R. Ramaswami and K.N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489–500, October 1995.

[95] 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*, vol. 18, no. 10, pp. 2062–2071, October 2000.

[96] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1061–1073, September 1998.

[97] G. Xiao and Y. Leung, "Algorithms for allocating wavelength converters in all-optical networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 545–557, August 1999.

[98] 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*, vol. 16, no. 5, pp. 729–735, May 1998.

[99] A. S. Acampora and I. A. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Transactions on Communications*, vol. 40, no. 6, pp. 1082–1090, June 1992.

[100] 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*, vol. 43, no. 1, pp. 88–98, January 1995.

[101] A. Bononi, G. A. Castanon, and O. K. Tonguz, "Analysis of hot-potato optical networks with wavelength conversion," *IEEE/OSA Journal of Lightwave Technology*, vol. 17, no. 4, pp. 525–534, April 1999.

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

[103] 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*, October 2000, pp. 235–246.

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

[105] 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.

[106] 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)*, April 2002, vol. 5, pp. 2673–2677.

[107] A. Detti, V. Eramo, and M. Listanti, "Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)," *IEEE/OSA Journal of Lightwave Technology*, vol. 20, no. 2, pp. 154–165, February 2002.

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

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

[110] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: Delay differentiation and packet scheduling," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 12–26, February 2002.

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

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

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

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

[115] C-H Loi, W. Liao, and D-N Yang, "Service differentiation in optical burst switched networks," in *Proceedings, IEEE Globecom*, November 2002, vol. 3, pp. 2313–2317.

[116] 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*, November 2002, vol. 3, pp. 2747–2751.

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

[118] J. Liu and N. Ansari, "Forward resource reservation for QoS provisioning in OBS systems," in *Proceedings, IEEE Globecom*, December 2003, vol. 3, pp. 2777–2781.

[119] 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)*, March 2002, pp. 774–775.

[120] 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.

[121] 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.

[122] 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)*, November 2001, p. TuD2.

[123] 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*, vol. 4, no. 6, pp. 68–80, November/December 2003.

[124] 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.

[125] Q. Zhang, V.M. Vokkarane, B. Chen, and J.P. Jue, "Early drop scheme for providing absolute QoS differentiation in optical burst-switched networks," in *Proceedings, IEEE Workshop on High Performance Switching and Routing (HPSR)*, June 2003, pp. 153–157.

[126] Q. Zhang, V. M. Vokkarane, B. Chen, and J. P. Jue, "Early drop and wavelength grouping schemes for providing absolute QoS differentiation in optical burst-switched networks," in *Proceedings, IEEE Globecom*, December 2003.

[127] Q. Zhang, V. M. Vokkarane, B. Chen, and J. P. Jue, "Absolute QoS differentiation in optical burst-switched networks," in *UTD Technical Report UTDCS-45-03*, October 2003.

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

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

[130] 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)*, March 2002, vol. WG6, pp. 221–222.

[131] A. Neukermans and R. Ramaswami, "Mems technology for optical networking applications," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 62–69, January 2001.

[132] T.-W. Yeow, K.L.E. Law, and A. Goldenberg, "Mems optical switches," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 158–163, November 2001.

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

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

[135] C. Hsu, T. Liu, and N. Huang, "Performance analysis of deflection routing in optical burst-switched networks," in *Proceedings, IEEE Infocom*, 2002, vol. 1, pp. 66–73.

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

[137] 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)*, May 2002, vol. 5, pp. 2678–2681.

[138] A. Detti, V. Eramo, and M. Listanti, "Optical burts switching with burst drop (OBS/BD): An easy OBS improvement," in *Proceedings, IEEE International Conference on Communications (ICC)*, May 2002.

[139] A. Maach and G.V. Bochmann, "Segmented burst switching: Enhancement of optical burst switching to decrease loss rate and support quality of service," in *Proceedings, Optical Network Design and Modeling (ONDM)*, 2002.

[140] Z. Rosberg, H. L. Vu, M. Zukerman, and J. White, "Blocking probabilities of optical burst switching networks based on reduced load fixed point approximations," in *Proceedings, IEEE Infocom*, 2003, vol. 3, pp. 2008–2018.

[141] M. Neuts, Z. Rosberg, H. L. Vu, J. White, and M. Zukerman, "The advantage of the burst segmentation option in optical burst switching," in *Proceedings, ICOCN 2002*, November 2002.

[142] M. Neuts, H. L. Vu, and M. Zukerman, "Insight into the benefit of burst segmentation in optical burst switching," in *Proceedings, COIN-PS 2002*, July 2002.

[143] H. L. Vu J. White M. Neuts, Z. Rosberg and M. Zukerman, "Performance enhancement of optical burst switching using burst segmentation," in *Proceedings, IEEE International Conference on Communications (ICC)*, May 2002, vol. 3, pp. 1828–1832.

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

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

[146] J. Ramamirtham and J. Turner, "Design of wavelength converting switches for optical burst switching," in *Proceedings, IEEE Infocom*, June 2002, pp. 2008–2018.

[147] 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)*, March 2002, pp. FD2–1–FD2–3.

[148] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 852–857, June 1996.

[149] R. Barry and P. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 858–867, June 1996.

[150] 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*, July 2002, vol. 4874, pp. 125–136.

[151] C. Qiao and M. Yoo, "Choices, features and issues in optical burst switching," *SPIE Optical Networks Magazine*, vol. 1, no. 2, pp. 36–44, 2000.

[152] 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.

[153] R. Jothi and V. M. Vokkarane, "Threshold-based differnetiated intermediate node initiated (TDINI) signaling: A hybrid reservation technique for optical burst-switched networks," in *Proceedings, 7th Informs TELECOM 2004 Conference*, March 2004.

[154] A. Detti and M. Listanti, "Impact of segments aggregation on TCP reno flows in optical burst switching networks," in *Proceedings, IEEE Infocom*, June 2002, vol. 3, pp. 1803–1812.

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

[156] S. Gowda, R. Shenai, K. Sivalingam, and H. C. Cankaya, "Performance evaluation of TCP over optical burst-switched (OBS) WDM networks," in *Proceedings, IEEE International Conference on Communications (ICC)*, May 2003, vol. 2, pp. 1433–1437.

[157] S.-Y. Wang, "Using TCP congestion control to improve the performances of optical burst switched networks," in *Proceedings, IEEE International Conference on Communications (ICC)*, May 2003, vol. 2, pp. 1438–1442.

[158] S.-H.G. Chan J.. He, "TCP and UDP performance for Internet over optical packet-switched networks," in *Proceedings, IEEE International Conference on Communications (ICC)*, May 2003, vol. 2, pp. 1350–1354.