CONTENTION RESOLUTION AND BURST GROOMING STRATEGIES IN

LAYERED OPTICAL BURST-SWITCHED NETWORKS

APPROVED BY SUPERVISORY COMMITTEE:

Dr. Jason P. Jue, Chair

Dr. Lakshman S. Tamil

Dr. Andrea Fumagalli

Dr. Murat Torlak

In the memory of

all brave low priority bursts that were discarded

because they believed in their destinations

and rejected any deflections,

in particular,

$B_C(h, e)$.

CONTENTION RESOLUTION AND BURST GROOMING STRATEGIES IN LAYERED OPTICAL BURST-SWITCHED NETWORKS

by

FARID FARAHMAND, B.S.E.E., M.S.E.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

August 2005

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Professor Jason P. Jue, for his guidance, support, and encouragement throughout my Ph.D. research. He has been a friend and a mentor, and his lessons in patience and positive attitude I will embrace for a life time.

I also like to express sincere appreciation to my committee members, Dr. Lakshman S. Tamil, Dr. Andrea Fumagalli, and Dr. Murat Torlak, for their valuable comments and feedbacks.

Throughout my research, I was fortunate to work with many great and talented individuals in the Advanced Networks Research Lab, all of whom contributed to my work and helped me with their insightful suggestions.

I owe the completion of this dissertation to the support, camaraderie, and assistance of my family and friends. To my parents, who crossed several continents, hoping that one day I become *someone*, and I doubt I ever fulfill their wish. To my brother, who, despite his youth, continues to be the *checkpoint* in every decision I make. To Grand Master Kim, who has been like a father to me and taught me the true meaning of indomitable spirit. To Ivonne, *mi inspiración*, who never doubted me. And to all my friends at Talar, who have been patiently listening to my countless grumbles and complaints, since I can remember.

July, 2005

CONTENTION RESOLUTION AND BURST GROOMING STRATEGIES IN

LAYERED OPTICAL BURST-SWITCHED NETWORKS

Publication No. _____

Farid Farahmand, Ph.D.
The University of Texas at Dallas, 2005

Supervising Professor: Dr. Jason P. Jue

The amount of raw bandwidth available on fiber optic links has increased dramatically with advances in dense wavelength division multiplexing (DWDM) technology. However, existing optical network architectures are unable to fully utilize this bandwidth to support future highly dynamic and bursty traffic. Optical burst switching (OBS) has been proposed as a new paradigm to achieve a practical balance between coarse-grained circuit switching and fine-grained packet switching, hence, better utilizing the available bandwidth.

In this dissertation, we analyze a number of issues involving the development of OBS technology, including reactive and proactive contention resolution mechanisms with service differentiation capability, hardware implementation of the scheduler, and data burst grooming. We also propose OBS as an alternative technology to support computationally intensive Grid applications.

A major problem in OBS networks is contention. We introduce a new approach called *Look-ahead Contention Resolution* to reduce packet loss in OBS networks, while supporting quality-of-service. We also propose a scalable hardware architecture, which can be used for implementing our developed contention resolution algorithm.

An alternative scheme to reduce contention is a proactive contention resolution mechanism. Our proposed feedback-based scheme can effectively improve network performance by adjusting the burst transmission rate at each node according to the network status.

An important issue in packet aggregation and generating data bursts in OBS networks is to reduce the padding overhead. Padding overhead is required when bursts are released before they reach their minimum length requirement. We introduce the concept of data burst grooming and develop two grooming algorithms to reduce padding overhead and thus, enhance network performance.

The evolution of OBS technology highly depends on its ability in supporting diverse applications. We introduce a general OBS framework, which can be implemented within the context of the layered Grid architecture.

We believe that the above contributions have addressed a number of fundamental issues facing practical development of OBS networks in order to be considered for future deployments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION TO OPTICAL NETWORKS

1.1 Introduction

The telecommunications industry has experienced extraordinary changes during the past 20 years. By the middle of the 90's, the infamous IP traffic curves were predicting an astonishing increase of 100 percent every five months. For example, in its Feb. 19, 1997 press release, WorldCom reported the traffic over the backbone almost doubling every quarter [1]. Similar claims continued to be made for a number of years and backed by government authorities. The former FCC Chairman, Reed Hundt, wrote that in 1999 data traffic was doubling every 90 days [2]. Such commonly accepted perceptions attracted huge amount of capital, yielding significant advances in telecommunications and networking technologies. On the other hand, the over-predictions of vast Internet growth and irrational investments caused a dramatic slowdown in the telecommunications industry [3]. In any case, the high-flying dotcom bubble of the 80's and 90's proved to be a passing phase and doomed to fizzle.[1]

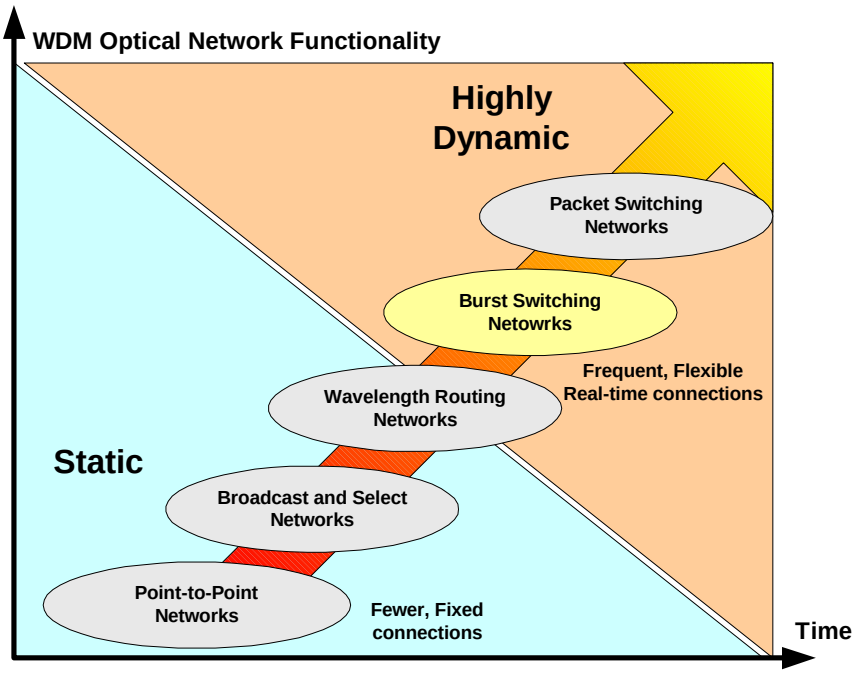
Today, the telecommunications industry is still struggling to overcome existing economic hurdles and hesitates to invest and deploy new technologies unless there is a sound potential for near-term return on investment. Yet, despite the industry crisis, network traffic is, in fact, growing steadily. Recent unbiased studies show that the number of Internet hosts continues to increase by 30 percent each year. This increase is estimated to result in about 70 percent growth in the number of connections [5]. In fact, traffic through the Internet is expected to increase by 50-100 percent within the next three to five years [6]. Such growth is fueled by a number of factors, including massive use of the World-Wide-

[1] Between 1999-2000, dotcoms spent $35 billion to build Internet-inspired communications networks and about 100 million miles of optical fiber (more than enough to reach the sun) were laid around the world. A year later, companies defaulted on $13.9 billion of telecommunications bonds, resulting in investor losses of $12.8 billion [4].

Web, growth in Voice-over-IP (VoIP), longer log times into the Internet using Web enabled PDAs and cellular phones, and heavy reliance of individuals on high-speed networks for their day-to-day operations. For example, consider the following facts: in 2003, the total world-wide e-commerce reached over $2 trillion [7]; in 2004, more than 664.5 million cell phones were sold [8]; in 2005, one billion users will be driving the Internet [9]; the total number of new wireless subscribers in 2004-2009 is expected to be 777.7 million, world-wide [10]; over 5,700 instant messages are exchanged every second (that is 15 billion messages every month) [11]. As more new data intensive applications, such as tele-medicine, e-Science, e-Astronomy [12], remote 3D graphics visualization, online multimedia conferencing, broadband residential services, are introduced, the demand for bandwidth continue to grow.

Optical technology have been considered as the logical choice to cope with such massive bandwidth growth. With theoretical available bandwidth of 25 tera bits per second (Tbps) per fiber in just the L-band window of operation, as well as low signal attenuation (0.2 dB/km), low signal distortion, low power requirement, and slow aging, optical networks will clearly play a key role in meeting existing and future demands. [13].[2] In order for optical networks to become fully applicable and support future on-demand high-speed applications at various network levels, optical technology must also offer reliability, transparency, simplicity, and scalability. Transporting a huge amount of data requires high reliability and consistency in performance. Transparency allows optical networks to be indifferent to the characteristics of the incoming data, such as its protocols or bit rate, and to simply *bypass* the incoming data. Simplicity of maintenance and hence, low operational cost, has widely been considered as the key factor in expanding optical networks. In addition, easy upgrade-ability within reasonable cost and robustness, remain to be critical issues in turning optical technology into the viable solution for future requirements.

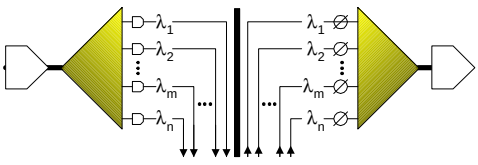
[2]One Tbps is equivalent to 200 million one-page emails or 35 million data connections at 28Kbps or 17 million digital voice telephone channel or half million compressed TV channels. An available bandwidth of 25 Tbps is about 1000 times the entire usable radio frequency (RF) spectrum on the planet earth (with its oxygen absorbtion at higher frequencies). [14].



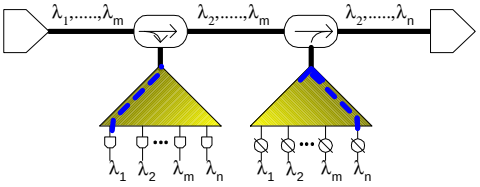
(a) WDM Network Evolution

Generation	First Generation	Second Generation	Optical Burst Switching	Optical Packet Switching
Components	ADM[c]	WADM [d]	OXC[e],[f]	OXC/WLC[f]
Toplogy	P2P WDM	Mesh/Ring	Mesh/Ring	Mesh
Capacity	10-100 Gb/s	100 Gb/s-Tb/s	100 Tb/s	Pb/s
Network Type	Opaque	Opaque/ Transparent	Transparent	Transparent
Switch Type	Opaque	Opaque/ Transparent	Transparent	Transparent
Traffic Type	Static	Lightpath-based/ Static/Dynamic	Burst-based/ Dynamic	Packet-based/ Dynamic

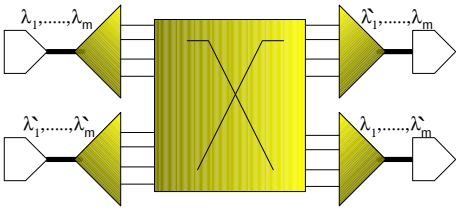
(b) WDM Network Characteristics



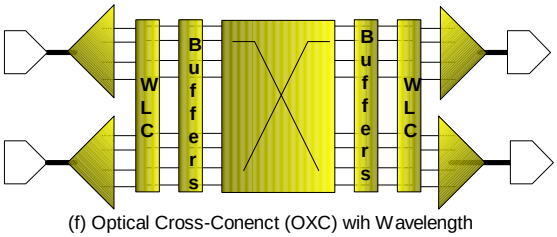
(c) Point-to-Point WDM Transmission



(d) Wavelength Add-and-Drop Multiplexing (WADM)



(e) Optical Cross-Conenct (OXC)



(f) Optical Cross-Conenct (OXC) wih Wavelength Converters (WLC) and Optical Buffers

Figure 1.1. Evolution of the WDM optical transport networks.

1.2 Wavelength Division Multiplexing (WDM)

A key technology in developing optical networks is Wavelength Division Multiplexing (WDM). WDM technology exploits the wide communication bandwidth in optical fiber. It enables each fiber to carry multiple optical signals, each at a different *wavelength*. In this way, WDM transforms a fiber into multiple *virtual fibers* [15]. Using WDM technology, it is possible to maintain low bit rate and multiply the number of wavelengths; this approach is particularly attractive to overcome technological challenges currently confronting 40 Gbps TDM systems (refer to [32] and its related references for more information on challenges involving development of 40 Gbps and faster TDM systems). Implementing WDM systems also results in reducing the number of required regenerators and hence, dramatically lowers the cost. As an example, consider transmitting a 40 Gbps signal over 600 km. Using a traditional system, this would require 16 separate fiber pairs with regenerators placed every 35 km for a total of 272 regenerators. A 16 channel WDM system, where each wavelength transmits at a rate of 2.5 Gbps, on the other hand, uses a single fiber pair and 4 amplifiers positioned every 120 km for a total of 600 km [16].

Wavelength multiplexing systems may be classified as either coarse or dense WDM, referred to as CWDM or DWDM, respectively, depending on their wavelength spacing and compliance with the International Telecommunication Union (ITU) industry standard wavelength grid. New generations of DWDM systems can support more wavelengths and thus, more channels over a single fiber link. A good summary of different types of WDM systems is provided in [17]. Today, many vendors are aiming to achieve about 1000 wavelengths per fiber. This can theoretically be achieved by utilizing a combined C and L-bands with spacing of 0.2 nm [33].[3]

[3]The C-band is from 1530 to 1579 nm and the L-band is from 1570 to 1610 nm.

1.3 Optical Network Classification

Given the growth of the optical community in recent years, it is important to develop a taxonomy for classes of optical networks. However, this has become a challenging proposition because depending on the available technologies, many terms and classifications are described differently. In the following sub-sections we classify optical networks as shown in Fig. 1.1(a) in the following order:

- First Generation Optical Networks;
- Second Generation Optical Networks;
- Optical Packet Switching Networks;
- Optical Burst Switching Networks.

Fig. 1.1(b) characterizes each class according to its topology, capacity, switching type, network characteristic, and the type of traffic it supports.

1.3.1 First Generation Optical Networks

The first generation optical network architecture only supports point-to-point configuration. That is, the entire traffic coming into each node will be converted from optics to electronics and terminated. Consequently, if all or a portion of the terminated traffic needs retransmission, the out going traffic will have to be converted into optics before being sent out. In point-to-point networks, each node must have full electronic add-and-drop multiplexing (ADM) capability, as shown in Fig. 1.1(c), which is costly and causes higher delay and possibly electronic bottleneck. Such a node architecture in which all incoming wavelength channels must be terminated and undergo electronic processing and switching is called *opaque*.

The inefficiency of point-to-point optical networks becomes particularly evident knowing that a typical node terminates only 30% of the incoming traffic and bypasses the

rest [18]. Examples of first generation optical networks are early SONET (synchronous optical network) and SDH (synchronous digital hierarchy) networks [19].

1.3.2 Second Generation Optical Networks

The second generation optical networks are often classified into two categories: *broadcast and select architecture* and *wavelength routing architecture* [20].

Broadcast and select networks use tunable transmitters and receivers and they transmit the signals to all other nodes using a *passive (active) star coupler*. Consequently, only the node with the appropriate receiver can detect the signal. Broadcast and select networks are classified as either single-hop or multihop. The terms single-hop and multihop indicate whether the data only traverses optical switching components on the end-to-end path (single-hop) or whether it traverses a combination of optical and electronic switching components (multihop). This is important because all-optical networks have the major advantage of bit rate transparency.

Wavelength-routed networks utilize wavelength add-and-drop multiplexing (WADM), Fig. 1.1(d), allowing the incoming traffic on each *wavelength* to be either passed through the node or dropped at the node. Therefore, WADM technology allows optical signals to be transmitted uninterrupted over longer distances, resulting in larger topological scopes. An important advantage of the second generation optical networks over their point-to-point counterpart is their cost efficiency, where the need for having electronic add-and-drop multiplexing and electronic processing at every node is eliminated. Clearly, optical amplifiers play an important role in successfully deploying second generation optical networks.

A major disadvantage of WADM, however, is that they are only statically configurable. That is, they could be used for optical circuits, known as *lightpaths*, which were carrying static traffic between two specific nodes. This is often referred to as *static lightpath routing*. Static wavelength routing are commonly used in SONET rings. In such networks, each node transmits on a specified wavelength and the receiving node must tune to a specific wavelength by means of wavelength-tunable lasers.

Wavelength-routed networks can also be implemented using optical cross-connects (OXC), shown in Fig. 1.1(e). A detailed architecture of a WDM node with optical cross-connect and electronic grooming capacity is provided in [21]. An OXC-based (or WADM-based) node is also called *translucent* node, which is transparent with respect to some of the optical data channels and opaque with respect to others.

The basic functionality of an optical cross-connect is to optically switch the incoming wavelengths on input (ingress) ports to wavelengths on the appropriate output (egress) ports. The OXC may be equipped with wavelength converters, in which case incoming light can change color before continuing to the next node. More sophisticated architectures allow waveband-switching in which a group of wavelengths can switch together [22].

By allowing fast tuning and switching, transparent OXCs can support *dynamic lightpath routing* and thus, satisfy on-demand requests under electronic control. Consequently, the network capacity can be enhanced and mesh as well as ring topologies can be supported.

Wavelength routed networks are categorized as optical circuit switching (OCS) networks. Optical circuit switching is supported by establishing static or dynamic lightpaths. Establishing lightpaths between nodes are suitable for constant rate traffic such as voice traffic, however, they may be unsuitable for highly dynamic traffic with short peaks. Furthermore, as lightpaths must be established using a two-way reservation scheme that incurs a round-trip delay, the high overhead of connection establishment may not be well-suited for short bursts of traffic. Under bursty traffic, sufficient bandwidth must be provisioned to support the peak traffic load, leading to inefficient network utilization at low or idle loads. A good discussion on economical impacts of accommodating all peak-traffic loads for Internet Service Providers (ISPs) is provided in [35].

1.3.3 Optical Packet Switching Networks

Optical packet switching (OPS) provides packet switched services at the optical layers. The goal of such networks is to provide the same services that electronic packet-switched networks, such as the Internet and ATM networks, but at much higher speed. This is

achieved by eliminating electronic switching and matching WDM transmission capacities. An optical cross-connect capable of supporting OPS network, including optical buffers and wavelength converters, is shown in Fig. 1.1(f).

In OPS networks, data is transmitted in form of optical packets [36]. These packets are transmitted across the optical core without having to be converted to electronics at intermediate core nodes. OPS can provide dynamic bandwidth allocation on a packet-by-packet basis. Such dynamic allocation leads to a high degree of statistical multiplexing, which enables the network to achieve a higher degree of utilization when the traffic is variable and bursty.

An example of an optical transport network is shown in Fig. 1.2. Such a network, typically, consists of a collection of edge nodes and core nodes connected to each other using WDM links. In this network, the traffic is often originated at the ingress node and terminated at one or more destination nodes called egress edge nodes. The ingress node receiving the incoming IP traffic from multiple client networks, such as SONET, ATM , or Gigabit Ethernet, and transmits it through high capacity DWDM links. Each core node is connected to one or more edge nodes, and depending on its capability, the core node can either pass-through the incoming optical signal to the next node or terminate it. After going through multiple core nodes, the optical signal eventually is terminated at the egress edge node. Upon receiving the data, the egress edge node sends the data to the corresponding client network.

In spite of advantages of OPS networks, given notable technical and pricing concerns, their full-scale deployment is far from reality. In the following paragraphs we briefly describe three major technical challenges in developing optical packet switching networks: optical switching, optical buffering, and optical packet synchronization. For more detailed information on enabling OPS technologies refer to [37]. A short description of optical building blocks is provided in the appendix at the end of this chapter.

In order to achieve packet switching, the switch (OXC) must have nanosecond switching time. Many different switching technologies have been developed and are being inves-

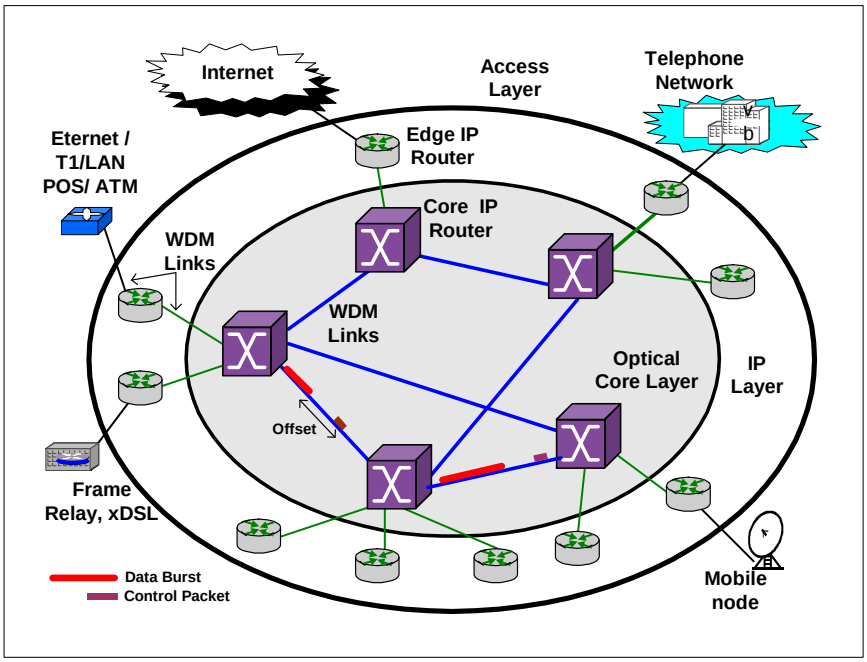


Figure 1.2. An optical network.

tigated in research laboratories. Examples of such technologies are Opto-mechanical optical switches, wave guide solid state optical switches (including electro-optical, thermal-optical, acousto-optical, and Liquid-crystal), micro-electro-mechanical optical switches (MEMS), integrated optics based switches (which are on silica , LiNbO3, polymer, GaAs, InP, and silicon materials), and bubble optical switches. The main requirements for any one of these technologies to be practical and widely applicable are their long term reliability, size, low optical loss and cross-talk radiation, scalability, fabrication, and cost. For example, in spite of their relative maturity, MEMS can only offer millisecond switching time. Hence, they cannot be considered for optical packet switching. On the other hand, LiNbO3-based optical switches offer nanosecond switching time but they have large insertion loss and they are very sensitive to fabrication imperfections. A summary of different switching technologies and their basic properties is available in [23].

The development of optical storage or memory has been considered as the key techno-

logical challenge in realizing the optical packet switching networks.[4] However, the current research in optical buffering and optical flip-flops is still in an early stage. A number of approaches have been proposes to mimic electronic flip-flops or RAMs in optical domain.

The basic idea behind optical flip-flop is to keep the optical data in optical format throughout the storage time without being converted into electronic format. Hence, the device will be able to turn on to store and off to release optical data at a very rapid rate by an external command. Many different technologies have been considered to store light. A majority of proposed architectures for optical flip-flops are based on the bistable operation of laser diodes and semiconductor optical amplifiers (SOAs) or coupling SOA-based MachZehnder interferometers (SOA-MZIs) [24]. Another approach to provide optical storage is slowing down the light pulses significantly. A possible way to achieve this is by using electromagnetically-induced transparency effect in quantum dots [25] and [26]. A more fundamental approach to develop optical buffering is to *trap* or *halt* photons in a coherent and reversible quantum state transfer between light and atoms, as proposed in [27] and [28].

Until optical storage devices are realized, optical fiber delays (FDLs), which are just long pieces of fiber, have been considered to temporarily buffer optical packets by simply *delaying* them.[5] Typically, about 1 km of fiber provides approximately 5 milliseconds of delay. Commercially available FDLs offer multi-milliseconds of delay with very small delay variations (less than 0.5 nanosecond) [29].

Optical packet synchronization involves two types of synchronization: timing and packet. Timing synchronization is required to time flip-flops which read the packet header frame at the bit level. Establishing timing synchronization requires clock transmission or timing extraction between nodes, and it must be acquired on a packet-by-packet basis unless an out-of-band signal is used to distribute the clock. The packet synchronization (or packet

[4] It is generally believed that an optical buffer would enable many other new optical systems such as optical signal processing, phase-arrayed antennas, and nonlinear optics.

[5] The process of packet delaying through FDLscan considerably reduce signal energy and in practice cannot be done indefinitely

delineation), on the other hand, refers to how fine we want to tune the position of each incoming packet before they are input into the switch. This is particularly important if packets arrive at some boundaries or the header packet is separated from its associated data packet in time, space, or both. In such cases, an input synchronization stage is implemented to align the incoming packets within the required boundaries [30].[6] Clearly, having smaller packet sizes require finer packet synchronization. However, if packets are long, the data-to-overhead ratio will be much larger, and hence, more guard times can be allowed between packets. This leads to less strict packet synchronization. Refer to [31] for a more detailed discussion on packet synchronization.

1.4 Optical Burst Switching Networks

Optical burst switched (OBS) [38], [39] has been proposed as a new paradigm to achieve a practical balance between coarse-grained circuit switching and fine-grained packet switching.[7] In OBS networks, incoming data is assembled into basic units, referred to as *data bursts* (DB), which are then transported over the optical core network. Control signaling is performed out-of-band by *control packets* (CP) which carry information such as the length, the destination address, and the QoS requirement of the optical burst. The control packet is separated from the burst by an offset time, which allows for the control packet to be processed at each intermediate node before the data burst arrives. Although, out-of-band signaling or separating the control and data packets in time are not inherent properties of OBS networks, in most literature they are conveniently assumed to be true. A number of works, however, have looked into such common assumptions. For example, [43] examines the implications of using offsets in OBS networks, while [44] proposes a variation of OBS

[6] Packet alignment requirement is primarily due to the fact that packet propagation speed varies with temperature and distance, with a typical figure of 40 $ps/^0C/km$. This implies that packets traveling through a 100-km of fiber under temperature variation range of 0 - 25 ^0C can experience a delay variation of 100 ns. This translates to 1000 bits when transmitting at 10 Gbps.

[7] The first formal introduction of *burst switching* was provided in early 80s in *Burst Switching an introduction*, by Amstuts [40] and later in [42] and [41]. Initially, this concept was considered as an extension to fast packet switching. The key idea in the proposed electronic burst switching was to handle packets of arbitrary length while employing decentralized shared buffer.

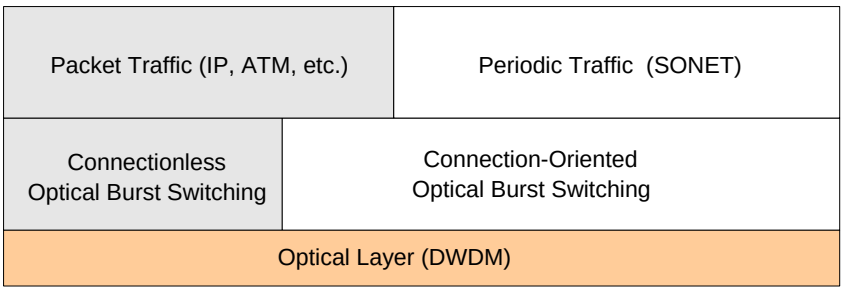


Figure 1.3. Supported services on optical burst switching networks.

without offset. A clear advantage of separating the header packets from their associated data bursts in time and space is that the header packets can be processed at slower speed electronically. However, a potential problem is packet synchronization, as we described before, which must be addressed.

By aggregating and providing out-of-band signalling, OBS provides dynamic bandwidth allocation and statistical multiplexing of data, while having fewer technological restrictions compared to OPS. For example, in OBS networks, optical buffering requirement can be eliminated or reduced. Due to having larger size packets (bursts, the readily available technologies such as FDLs can provide the limited buffering required by OBS networks. Furthermore, because packet aggregation in OBS networks increases the data-to-overhead ratio, packet synchronization requirements are less stringent and can be performed at lower speeds. Packet aggregation in OBS networks, on the other hand, results in potentially higher end-to-end delay and high packet loss per contention, due to lack of sufficient buffering.

Optical burst switching networks can support different networking modes. The networking mode is primarily either *connection-oriented* or *connectionless*. Connection-oriented networks are those in which the connection setup is performed prior to information transfer. In contrast, in connectionless networks no explicit connection setup actions are performed prior to transmitting data; instead data packets are routed to their destinations based on information in their header. Fig. 1.3 shows different examples of connection-

Optical Transport Networks	Bandwidth Utilization	Traffic Adaptability	Latency (set-up)	Over head	Optical Buffer Requirements	Data Loss
Optical Circuit Switching	Low	Low	High	Low	None	Low
Optical Packet Switching	High	High	Low	High	High	Low
Optical Burst Switching	High	High	Low	Low	Low	High

Figure 1.4. Comparing different optical switching technologies.

oriented and connectionless networks. We consider ATM or SONET networks as examples of connection-oriented networks. On the other hand, we consider an IP networks as an examples of connection-less network. However, IP networks with Resource Reservation Protocol (RSVP) and/or multiprotocol label switching (MPLS) mode of operations are recognized as connection-oriented networks.

In order to support connection-oriented services on OBS, a two-way reservation protocol, such as tell-and-wait (TAW) can reserve the end-to-end path for the requested duration, prior to data transmission. Connectionless services on OBS can be supported by various one-way reservation protocols, such as tell-and-go (TAG) and just-enough-time (JET) [38]. In this document, we will focus on the connectionless mode of operation of OBS; however, the framework for the control plane will be general enough to support any out-of-band signaling scheme.

Fig. 1.4 summarizes the characteristics of optical circuit switching (supported by wavelength-routed networks), optical packet switching, and optical burst switching. As indicated in this figure, the main advantages of OBS technology is its low requirement for optical buffering and low average setup latency. Although the burst latency setup is low, since packets must be delayed until the burst is ready to be transmitted, on average, packets experience longer average end-to-end delay when compared to packet switching.

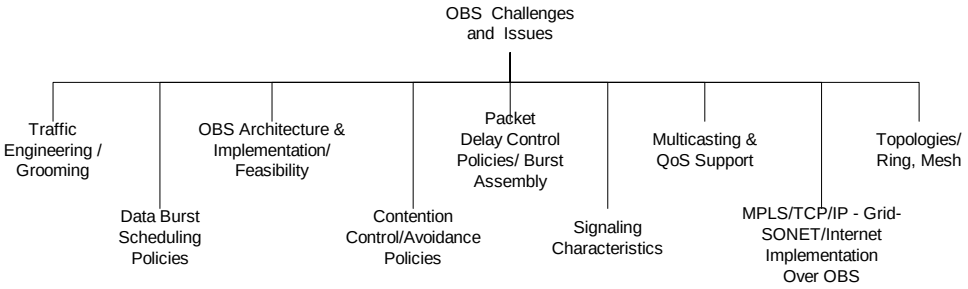


Figure 1.5. OBS technologies and challenges.

Furthermore, OBS tends to reduce the total overhead as well as the processing power requirement. These are mainly due to the fact that fewer individual packets are transmitted in OBS for the same number of incoming IP packets. On the other hand, the main concern in OBS networks is high loss rate. A practical approach to reduce high rate of loss in OBS networks is by using fiber delay lines. However, the tradeoff will be high cost and complexity.

When multiple number of packets with similar characteristics, such as edge node destination, quality-of-service, etc., are assembled into a single burst and at the same time the usage of optical buffering is eliminated or minimized, new effective techniques are required to reduce IP packet blocking or average end-to-end IP packet delay due to packet aggregation. Consequently, issues, such as contention resolution, quality-of-service, supporting TCP-layer, etc., become important issues which require close attention in OBS networks. Fig. 1.5 summarizes some of these issues.

Aside from technical challenges, evolution of OBS technology highly depends on its ability in supporting diverse applications. Many researchers have been investigating the implementation of OBS technology to support applications such as Grid computing and distributed database. Clearly, the main criteria in supporting such applications is that they must be able to tolerate a degree of delay and loss.

1.5 Organization of Dissertation

This dissertation consists of eight chapters. In this chapter we outlined the basic properties of optical burst switching technology and how it is compared with other optical switching technologies. Chapter 2 focuses on basic components in an OBS network and the architecture of each. In Chapter 2, we also briefly examine the latest developments pertaining to optical burst switching particularly in areas of supporting quality-of-service, burst assembly, and contention resolution mechanisms. Chapter 3 presents a layered view of OBS protocols, separating them into data and control planes. Chapter 4 examines a number of existing contention resolution mechanisms in OBS networks. We introduce two new contention resolution algorithms and compare their performance with well-known existing algorithms. In Chapter 5, a rate-based contention avoidance mechanism is introduced to reduce packet congestion in OBS networks. Chapter 6 addresses the data burst grooming and provides an edge node architecture enabling the data burst grooming capacity. In Chapter 7, a layered architecture for Grid-over-OBS is presented and we position OBS protocol stack within the framework of the layered Grid architecture. We describe how different layers of the Grid interact with OBS layers and elaborate on protocols supported by each layer. Chapter 8 concludes this dissertation and identifies some possible areas for future research.

1.6 Appendix A: Basic Optical Components

Development of optical networks cannot be fully understood without having a basic knowledge about its key building blocks. We briefly name the major optical components used in optical networks and describe their basic functionalities.

Couplers: These devices combine light into fiber or split light out of a fiber. Three common type of couplers are splitters, combiners, and directional couplers.

Optical fiber: Optical fibers are essential building blocks of any optical transmission system. Typical fiber characteristics include low insertion losses, low wavelength shift, and low cross talk from adjacent signals.

Optical amplifiers: These devices play an important part in optical transmission systems. Optical signals are prone to losses in the fibers as they propagate through them. These signals have to be strengthened to enable propagation. Optical amplifiers are used in three different ways in a fiber transmission system: power amplifier, line amplifier, preamplifiers. Depending on the fiber type, and the distance between the transmitter and receiver, different types of optical amplifiers may be needed. Common types of amplifiers include Erbium-Doped Fiber Amplifier (EDFA), Semiconductor Optical Amplifier (SOA), and Raman Amplifier.

Transmitters and receivers: The basic operation of these devices is converting digital signals into optical signals or converting optical signals to digital signals, respectively. Transmitters differ depending on the type of the lasers they use. Examples of laser types include Semiconductor Laser Diodes, Fabry-Perot Lasers, External Cavity Laser, and Mechanically Tuned Lasers. An important characteristic of an optical receiver is its sensitivity toward the received optical signal.

Switches: Switches are the vital components in any optical networks. Switches allow optical signals to be switched without having to convert them to electronic signals. Different types of switches can be employed in optical networks such as Fiber cross-connects, wavelength-routing switches, and photonic packet switches.

Wavelength converters: The function of wavelength converters is to convert data from the incoming wavelength to an outgoing wavelength. Classification of wavelength converters is done based on the wavelength range they operate. The basic types of wavelength converters are fixed-input/fixed-output, variable-input/fixed-output, fixed-input/variable-output, variable-input/variable-output.

CHAPTER 2
OPTICAL BURST SWITCHING ARCHITECTURE

2.1 Introduction

Optical burst switching (OBS) has been proposed as an efficient way to satisfy future on-demand applications with high bandwidth requirements [38]. In an IP-centric OBS, IP packets are assembled into super-size packets called data bursts. These bursts are transmitted following a burst header packet (BHP) after some offset time [45]. Each BHP contains routing, scheduling, and packet priority information and is processed electronically prior to its data burst arrival. Consequently, when the data burst arrives, it can ”cut-through” the switch on the pre-assigned path with minimum processing. Different signaling and scheduling mechanisms for reserving and releasing resources have been proposed for OBS.

Basic components of an OBS network include edge nodes, core nodes, and WDM links connecting the nodes together. In the remainder of this chapter, first, in Section 2.2, we describe the architecture of different components in the OBS network, including the edge and cores. Then, in Section 2.3, we briefly examine the latest developments pertaining optical burst switching, such as supporting quality-of-service, burst assembly, and contention resolution mechanism. We conclude the chapter in Section 2.4.

2.2 OBS Architecture

Fig. 2.1 shows an OBS network with DWDM links. The OBS network consists of a set of edge and core nodes. The traffic from multiple client networks is accumulated at the ingress edge nodes and transmitted through high capacity DWDM links over the core. Edge nodes provide legacy interfaces such as Gigabit Ethernet, ATM, and IP, and are responsible for data burst assembly and disassembly. Data burst assembly refers to the process by which incoming packets are aggregated into data bursts. The reverse operation is referred to as the disassembly process.

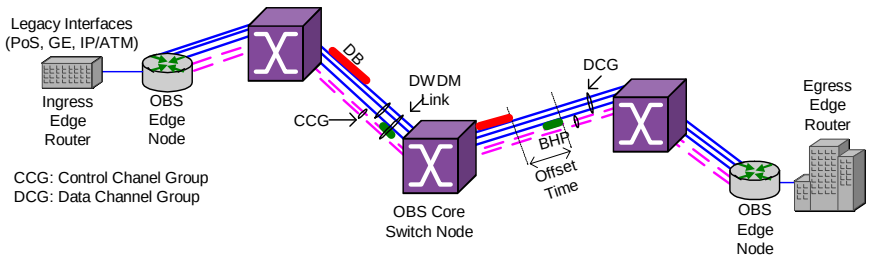


Figure 2.1. Optical burst-switched network.

When a data burst is ready to be transmitted, it is sent out following its header packet with some offset time. Data bursts are transmitted on dedicated sets of channels called Data Channel Group (DCG). On the other hand, The burst header packets (BHP) corresponding to the bursts are transmitted on a dedicated set of channels called Control Channel Group (CCG).

2.2.1 Slotted and Unslotted OBS Networks

OBS networks can be divided into two broad categories: *slotted* and *unslotted*. In synchronous slotted OBS networks, as shown in Fig. 2.2(a), data bursts and BHPs are only transmitted on their slot boundaries. [1] In this transmission scheme, control and data channels are divided into time slots with fixed duration. Each control slot is further divided into several BHP slots with fixed duration. When a data burst is ready for transmission, its header packet is first transmitted on an available BHP slot on the control channel. After an offset time, at the start of a new data slot, the associated data burst is transmitted. It is therefore convenient to represent offset time and burst duration in terms of slots. Note that data bursts in slotted transmission can have variable or fixed durations.

Fig. 2.2(b) shows an asynchronous unslotted transmission of data bursts and BHPs. In such a network there is no need to delay a data burst and its BHP until the appropriate slot boundaries have arrived. Although data bursts and their BHPs can be transmitted at any

[1] Many authors constrain slotted transmission to a case where all packets have the same length. In OBS slotted transmission, however, we assume bursts can have variable lengths.

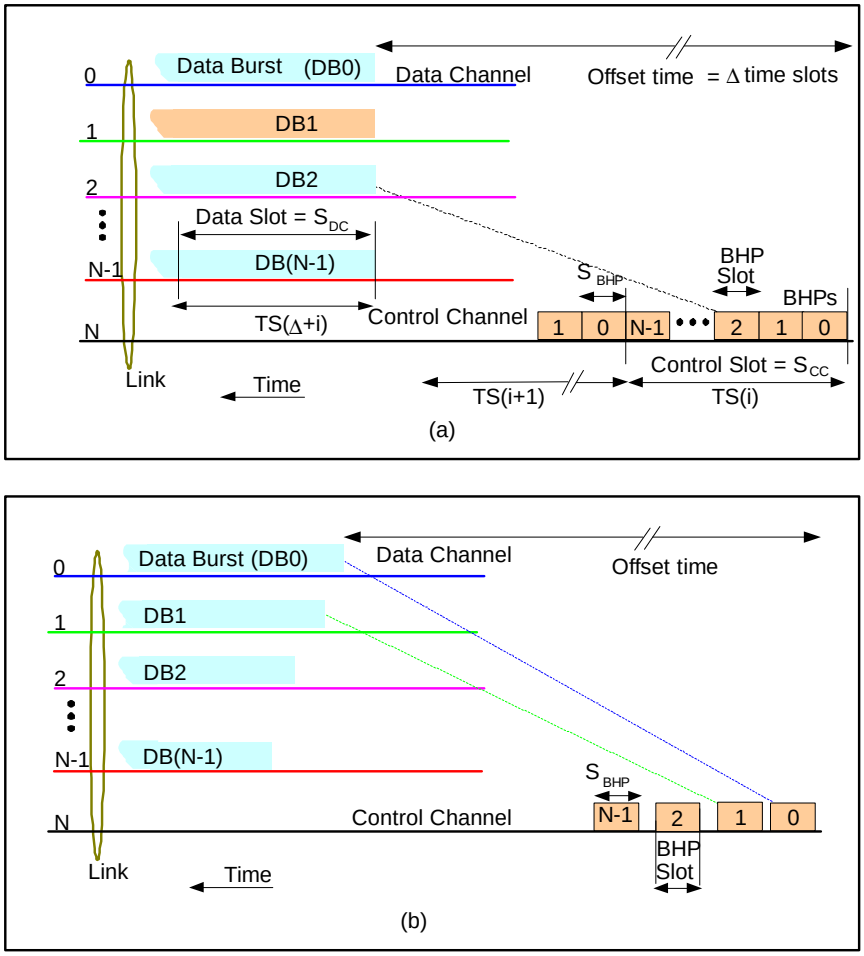


Figure 2.2. Data bursts and their BHPs in (a) the synchronous slotted and (b) the asynchronous unslotted transmission networks.

time, the start and end of the BHP's associated data burst must still be identified in terms of some predefined time units. Even though, in theory, data bursts in unslotted OBS networks can have fixed or variable lengths, for practical reasons, we only consider the latter case.

In general, since data bursts in slotted networks are transmitted in discrete sizes, bandwidth efficiency (the ratio between the number of data bytes over the total number of data and overhead bytes) [46] reduces at low traffic loads because data slots are not fully utilized. On the other hand, because arbitrary time units are used to represent data burst duration and offset time, in unslotted OBS networks the size of the header packet may be longer. In addition, the lack of slot boundaries in unslotted networks, eliminates synchronization time which may lead to lower average end-to-end packet delay.

In terms of data burst loss rate, simulation results for photonic packet switching indicate that the performance of slotted transmission with fixed-size packet length is superior to an unslotted system with variable size packets [47]. This is assuming that the average packet duration is equivalent to the fixed-size packet. Having smaller size data bursts can lower the loss rate. Clearly, the tradeoff to this is lowering the bandwidth efficiency.

The implementation complexity of slotted and unslotted packet switching has been described in [31]. However, in OBS systems, the unslotted transmission mechanism involves some additional complexities. This is mainly due to fact that the core switch nodes must resynchronize and align the data bursts and their associated BHPs.

A quick comparison between slotted OBS networks with fixed and variable size data bursts shows that the former results in lower bandwidth efficiency whereas the latter is slightly more complex to construct and requires more information fields in the header packet. The average end-to-end IP packet delay and the loss rate in variable and fixed sized slotted OBS depends on a number of factors, including the input traffic characteristics, load, and the edge node criteria for determining when bursts can be released. For example, at the edge node, the data bursts can be transmitted when a time threshold has been reached or when a specific length requirement has been met [48], [49].

2.2.2 Edge Node Architecture

A flexible edge node is expected to support various interfaces and aggregate the incoming packets into bursts. Fig. 2.3 shows the basic four modules used in the edge node architecture, namely, Line, Switch, Burst and Optical Assembly modules. We describe the basic characteristics of these modules in ingress and egress edge nodes.

The line modules (LM) provide the interface between the optical burst-switched network and other legacy networks including, ATM, IP, or even SONET rings. The primary function of LM is to convert different line interfaces into a common protocol (such as IP interface) for the switch module.[2] Typically, LMs in an edge node are add-on modules,

[2]This is referred as IP-centric OBS network which has been discussed in most literature. However, it is

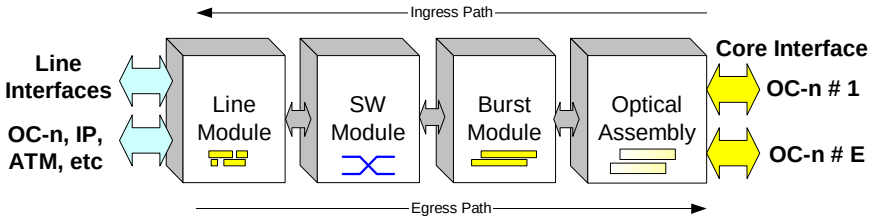


Figure 2.3. Basic edge node modules.

which can be provisioned as new applications are evolved. In case the edge node is required to support a connection-oriented service, the line module can be transparent to the incoming interface.

The main function of the switch module (SM) is to route IP packets into the proper burst module. Such routing mechanism can depend on a number of criteria, such as OBS destination of IP packets, or IP packet priority. This routing function is performed entirely in the electrical domain. The size of the switch depends on the number of egress ports on each node, E, number of channels on each egress port, N, number of supported service types, Q, and the number of line modules, I.

The burst module (BM) is responsible for assembling IP packets into bursts and decomposing incoming bursts from the egress direction into IP packets. In addition, it is responsible for determining when the burst must be released as well as burst scheduling on an available channel. The number of egress and ingress interfaces, E and I respectively, are independent parameters.

Fig. 2.4 provides a more detailed description of an ingress edge node architecture and its structural blocks. The front-end interfaces to the edge node provided by line modules can be electrical or optical in order to support varieties of services. All optical signals are converted to electrical prior to processing. The main function of the header reader block is to determine the destination of the incoming packet and the type of service it requires. This information is passed on to the node's control unit and eventually to the switch scheduler

possible to envision OBS networks which are based on an application layer other than IP.

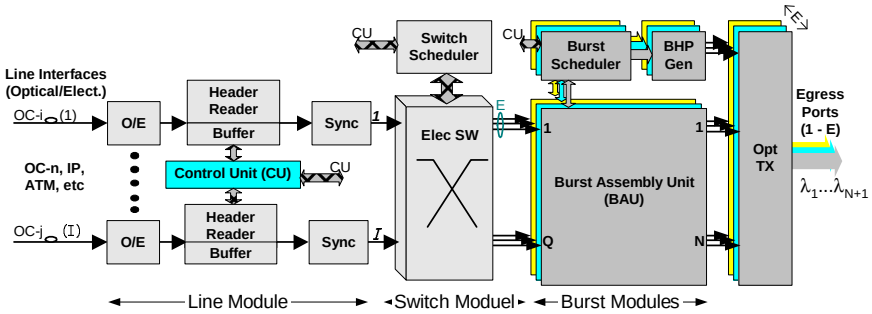


Figure 2.4. Ingress edge node architecture.

block. Prior to sending the packets into the switch, packet synchronization may be required depending on the transmission and switching technologies. The switch module forwards the IP packet into one of the E burst modules, each of which is connected to an egress port. Therefore, the switch module will have as many as $E \cdot Q$ egress ports and each edge node supports Q service types.

The burst module consists of the following units: burst assembly (BAU), burst scheduler (BSU), and BHP generator (BHPGen). Fig. 2.5 shows details of the BAU. Functionally, this unit operates as a virtual output queue. The incoming IP packets are directed into one of the burst formation queues. There will be as many as Q burst formation queues in each BAU and the size of each must be equal to the largest possible burst size. Broadly speaking, the burst scheduler unit can provide four basic functionalities: (a) defining the burst formation criteria, including the burst size and when the burst is ready to be transmitted; (b) scheduling the burst on an available channel for transmission; (c) transmission smoothing, which is used to condition the traffic and avoid network congestion; and (d) possible retransmission of bursts, which will require extra storage capacity.

Fig. 2.6 shows the optical assembly module. Assuming there is a single control channel ($M = 1$), all BHPs are transmitted on a dedicated wavelength. The remaining N channels can be assigned to data bursts. Each burst is optically modulated and passed on to the fiber link on a specific wavelength. The WDM unit multiplexes all wavelengths and transmit them to the next hop.

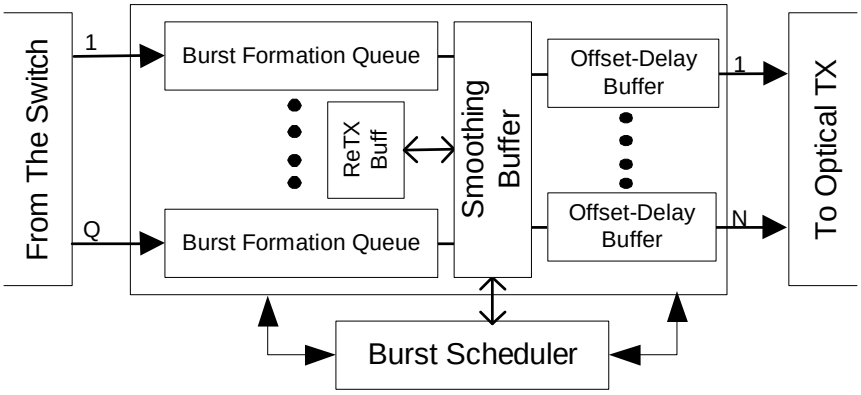


Figure 2.5. Burst assembly unit (BAU).

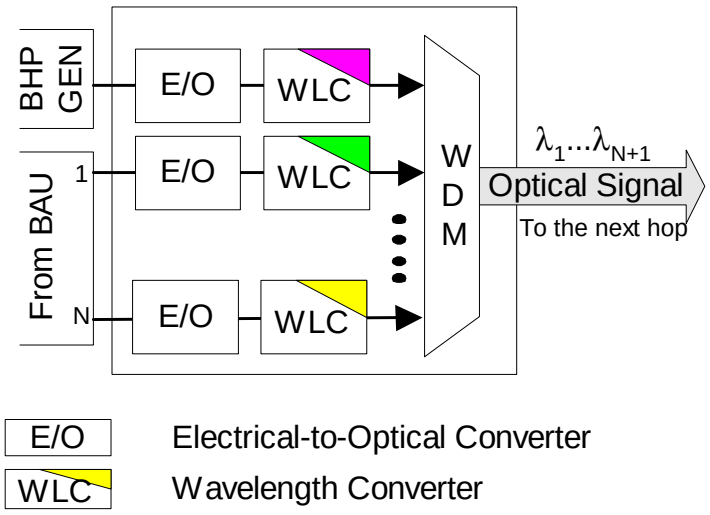


Figure 2.6. Optical assembly module in the edge node.

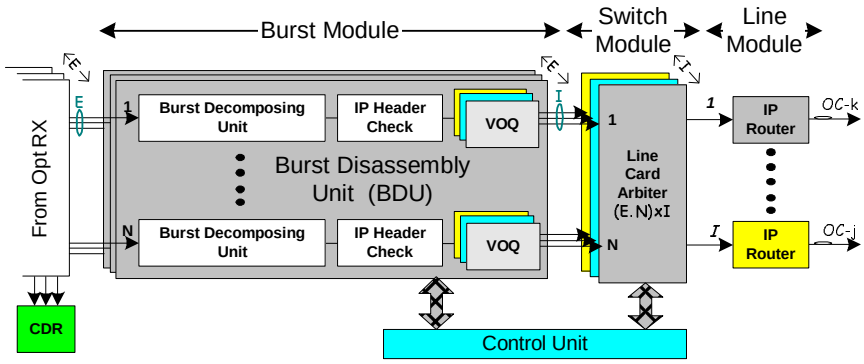


Figure 2.7. Egress edge node architecture

The operation of an edge node is shown in Fig. 2.7. In this case, all BHPs and data bursts are terminated and converted from optical into electrical signals. The clock data recovery (CDR) circuit can select, recover, and lock to one of the incoming egress links and provide the timing for the module. A burst disassembly unit (BDU) recovers IP packets from the incoming data burst frame. It initially decomposes each burst packet and then disassembles the burst packet by extracting and buffering individual IP packets. This is performed by the burst decomposing unit. In case of detecting a frame error, a burst retransmission request can be sent to the source edge node or IP client network. Then, the each IP header packet is read and sent to the appropriate virtual output queue (VOQ).[3] The output of each VOQ is directed to one of line card arbiters (LCA), where IP traffic is distributed between different IP routers, as shown in Fig. 2.7. Complex algorithms can be designed for LCAs to resolve QoS issues and support different types of services.

The above generic architecture encounters major technological challenges including fast processing, memory management, and buffer sizes. Fast processing is critical to reduce the size of buffers in the edge node. Hardware based algorithms in line card arbiters, such as Binary Tree [51], switch scheduler unit, or burst scheduler unit are essential in order to make fast decisions, resolve contentions, and packet routing. On the other hand, considering the volume of information an edge node is expected to carry, memory manage-

[3] A VOQ is dedicated FIFO for each line module.

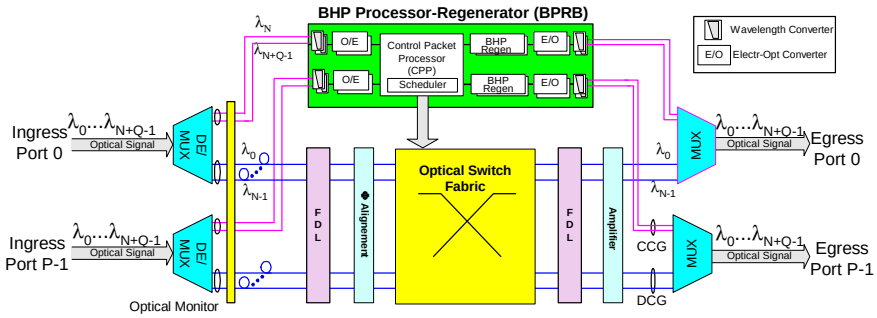


Figure 2.8. Typical architecture of the OBS core switch node with optical burst alignment capacity.

ment imposes a major issue. As an example, consider an edge node with 16 burst modules each having a bandwidth capacity of 10 Gbps. In this case, the system must be able to handle 160 Gigabits of data per second. This cannot be achieved without massive advanced parallelism to reduce instruction times. However, more parallelism implies larger buffer size requirements. Traffic fluctuation due to bursty nature of Internet traffic, inflicts even greater concerns regarding the size of buffers.

2.2.3 Core Node Architecture

A general description of edge nodes in OBS networks has been provided in [100]. In this subsection we briefly describe the overall core switch node architecture. Fig. 2.8 shows the generic core switch node architecture in OBS. In this hybrid architecture the core switch is fully transparent to optical data bursts, while the control channels are converted into electrical signals. Each ingress link is initially demultiplexed and all data and control channels are separated. Individual optical channels are examined for optical characteristics, including the optical power level and signal-to-noise ratio.

Fiber delay line (FDL) blocks can be used for variety of reasons. One potential application is to use FDLs as input or output optical buffers to delay data bursts when multiple bursts are contending for the same egress port in the switch fabric. Another possible application of FDLs is to compensate for BHP processing time delay in which data bursts are deliberately delayed in order to maintain the necessary offset times, as shown in Fig. 2.9. In slotted transmission, before data bursts go through the switch fabric, they must be

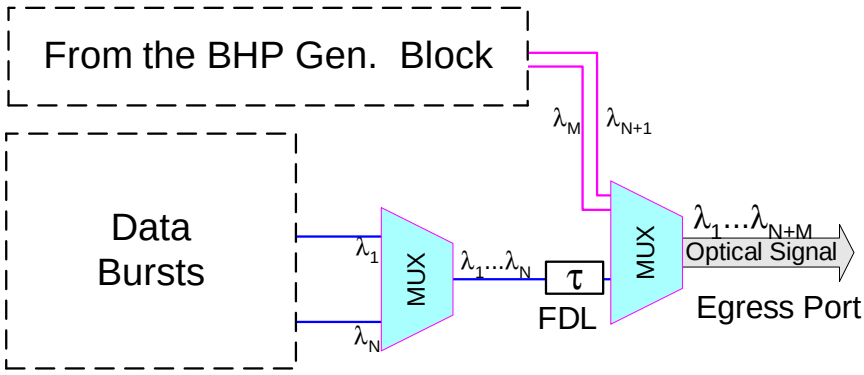


Figure 2.9. Using FDLs to compensate for BHP processing time delay.

aligned to time slots [50]. This operation can be performed using incremental fiber delay blocks, tunable converters, or more exotic approaches such as optical phase locked loops.

Many researchers have proposed various switch fabric architectures [52], [53]. An important issue in the switch fabric design is its cost and scalability [54]. In order to improve performance, many switch fabric architectures include wavelength converters. The optical couplers and wavelength converters in the switch fabric, along with the FDL blocks all cause optical loss of energy on the outgoing signals. Therefore, use of optical amplifiers prior to data burst transmission may be required. A typical switch fabric architecture is shown in Fig. 2.10

Incoming BHPs on control channels are processed and regenerated in the BHP processor-regenerator block (BPRB). In the BPRB the BHPs are first converted into electrical signals and then sent to a control packet processor (CPP), where they are processed and scheduled if proper resources are available. If a BHP request was successfully reserved, the switch fabric setup needs to be updated as the corresponding data burst arrives and leaves the switch. Furthermore, each accepted BHP must be regenerated with the updated information and transmitted to the downstream core node. The control packet processor is considered as the main part of the core node's BPRB and contains the data burst reservation algorithm.

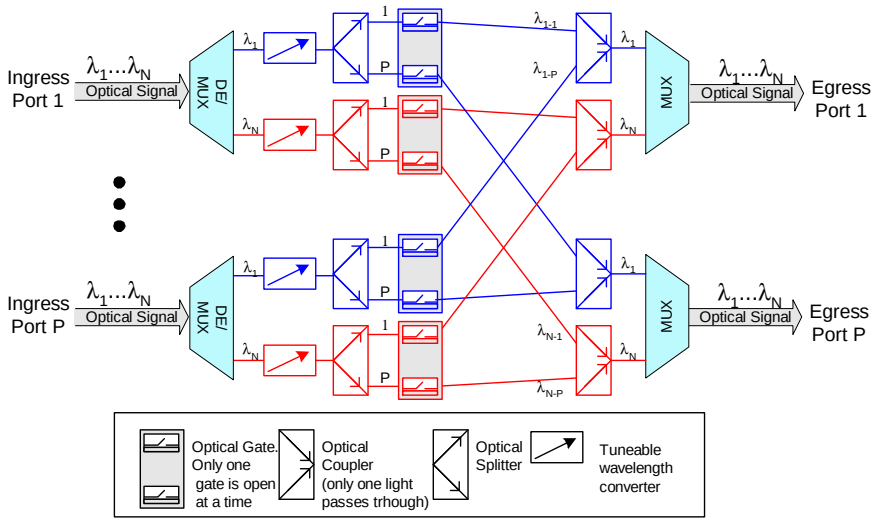


Figure 2.10. Core node's switch fabric.

2.2.4 Control Packet Processor

In this subsection we describe details of the CPP shown in Fig. 2.8. Fig. 2.11 demonstrates two different design architectures for the CPP block in the core node: *centralized* and *distributed*. In the centralized (pipelined) architecture, as shown in Fig. 2.11(a), each BHP is initially received by the receiver block, and its payload, including data burst length, destination, offset, QoS, is extracted. Then, the reformatted BHP request will be stored in the priority queue in which requests with higher QoS are given service priority. The scheduler block processes individual requests from each receiver queue based on their destinations. Upon acceptance of the request, the reservation is stored in the scheduler block until its associated data burst is serviced. The switch control block provides an interface between the scheduler and the switch fabric, updating ingress- egress channel connections.

In the distributed (parallel) architecture of the CPP, as shown in Fig. 2.11(b), egress ports have their own independent scheduler blocks. Each incoming BHP is decoded and checked for its destination. Then, the BHP is forwarded to one of the P destination queues connected to the BHP receiver block. Destination queues with similar index numbers are interfaced to the same scheduler block. The scheduler block processes the request, and upon making a reservation, the reservation will be stored until its associated data burst is

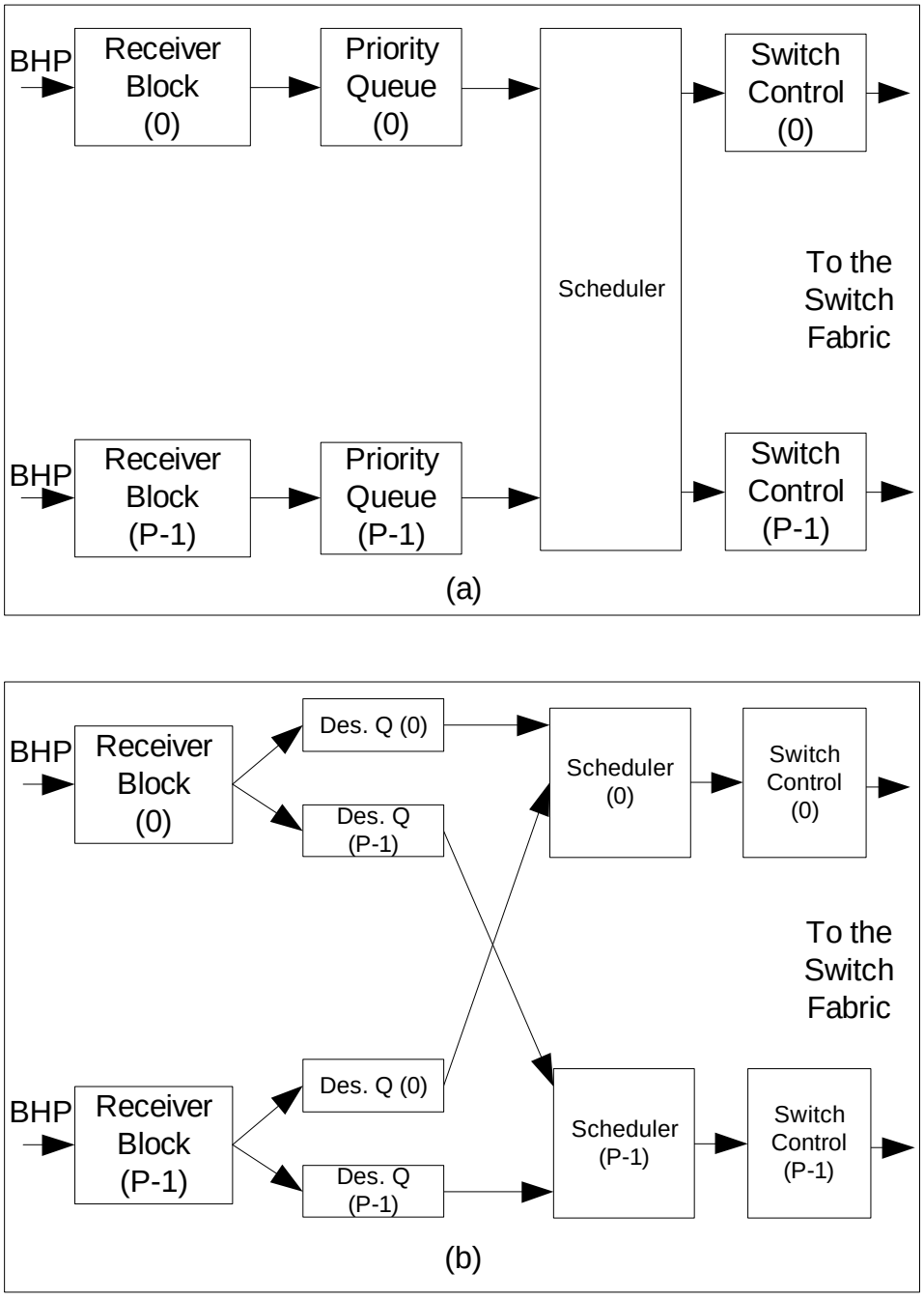


Figure 2.11. Control packet processor (CPP) architectures: (a) centralized (pipelined) and (b) distributed (parallel).

serviced.

The centralized and distributed architectures of the CPP, to a large extent, resemble input queuing and virtual output queuing systems [55], respectively. In the distributed architecture, by arranging input buffers according to egress ports, we can achieve parallel processing of BHP requests with different destinations. The distributed architecture also minimizes the problem of head-of-queue blocking. Furthermore, the distributed architecture is more reliable and provides better scalability in terms of modularity and adding new control channels. However, one important disadvantage of the distributed scheme is its high relative memory requirement and RAM usage. Assuming P represents the number of control channels entering the CPP, there will be P^2 destination queues, each of which must be dimensioned for the worst-case traffic condition. This is P times more than the total number of priority queues used in the centralized scheme.

2.3 OBS Issues and Challenges

In this section we provide a brief summary of various protocols dealing with critical issues in OBS networks, namely contention resolution and quality-of-Service. We also describe some of the existing protocols addressing and evaluating TCP over OBS and how efficiently OBS can handle TCP-based applications. We conclude this section by briefly discussing some of the practical applications proposed for OBS technology.

2.3.1 Contention Resolution Schemes

A major issue in OBS networks is contention. Contention resolution schemes can be categorized into reactive and proactive approaches, as shown in Fig. 2.12. Reactive approaches are provoked after contention occurs. Examples of reactive contention resolution schemes are space deflection (such as deflection routing), time deflection (such as buffering and delaying the data), wavelength conversion, and soft-contention resolution policies [62], [72], [90]. When one or more bursts must be dropped, the policy for selecting which bursts to drop is referred to as the soft contention resolution policy. Several soft con-

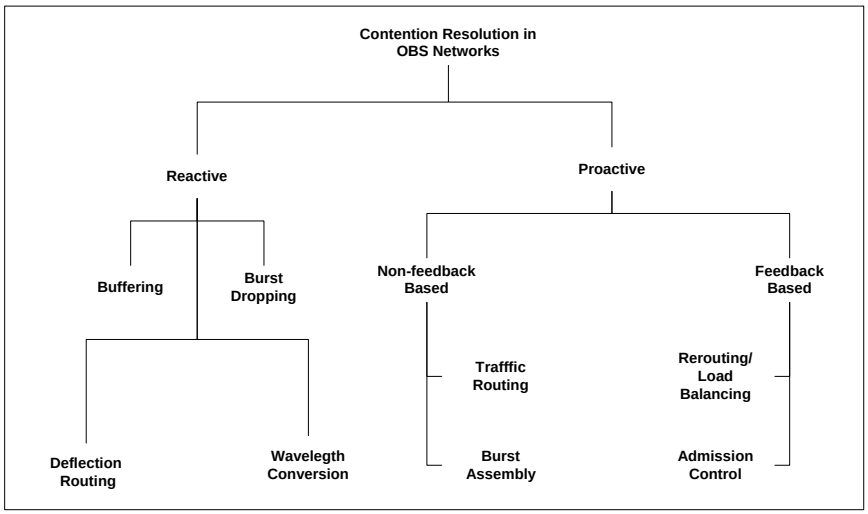


Figure 2.12. Classification of different contention resolutions.

tention resolution algorithms have been proposed and studied in earlier literature, including the shortest-drop policy [68], segmentation [92], and look-ahead contention resolution [67]. In proactive contention resolution approaches, traffic management policies are invoked to prevent the network entering the congestion state. Such schemes can be classified as non feedback-based or feedback-based. In a non feedback-based scheme, the ingress nodes have no knowledge of the network state and they cannot respond to changes in the network load. This can be achieved through traffic load balancing or data burst assembly.

In a feedback-based scheme, contention avoidance is achieved by dynamically varying the data burst flows at the source to match the latest status of the network and its available resources. One way to achieve this is to reroute some of the traffic from heavily loaded paths to under-utilized paths [70]. A similar approach has also been introduced by [73] where the authors consider balancing the data burst traffic between predefined alternative paths. In [71] a global load-balancing contention resolution scheme is proposed and its performance is examined for both dynamic and static traffic. Another way to avoid contention is to implement a TCP-like congestion avoidance mechanism to regulate the burst transmission rate [60], [95], [91]. In this approach, the ingress edge nodes receive TCP ACK packets from egress edge nodes, calculate the most congested links, and reroute their

traffic accordingly. In [97] burst cloning is proposed as an effective way to reduce data burst loss. The basic idea in burst cloning is to replicate a burst at appropriate nodes and send duplicated copies of the burst through the network simultaneously. In [66] a feedback-based OBS network is proposed in which using explicit feedback signaling to each source, the required data burst flow rate going to congested links is controlled. Clearly, a major concern with having feedback-based proactive contention resolution schemes is additional signaling overhead and signals processing. Hence, it is critical to design signaling protocols which are simple to implement and require minimum overhead.

2.3.2 Quality-of-Service

Quality-of-Service (QoS) in the Internet is critical due to service requirements needed by different applications. Hence, an important issue in OBS networks is supporting QoS. Quality-of-service schemes can be implemented in conjunction with existing contention resolution mechanisms and scheduling algorithms. Such schemes can be based on providing loss, delay, or bandwidth constraints or differentiation. Clearly, two important objectives in any QoS model are to ensure fairness and maintain high utilization.

Broadly speaking, regardless of the metric parameter, QoS schemes are classified as relative and absolute methods. In the relative QoS model, the performance of each class is defined relative to other classes. In such methods, there is no upper bound guarantee on the high priority-class loss probability. Several schemes have been developed to support the relative QoS model. For example, in offset-based QoS, extra offset is given to data bursts with higher priority resulting them to have relatively lower overall blocking probability. This scheme, known as prioritized JET, is proposed in [80] and its limitations are discussed in [69], [75], [99]. In [99] a proportional QoS scheme based on per-hop information is proposed. In this case, in order to maintain the differentiation loss factor between different classes, an intentional burst dropping scheme is employed. In [75], a proportional bandwidth scheme is used in parallel with policing on the burst assembly mechanism and with FDL buffering. In [94] relative QoS is provided by maintaining the number of wavelengths

occupied by each class of bursts. In this scheme, each class of service has a preset usage ratio of available bandwidth. Incoming bursts which are under-utilizing their share can preempt data bursts violating their assigned share.

The absolute QoS (or quantitative QoS), on the other hand, provides a bound guarantee for the desired traffic metric such as loss probability of different classes. Typically, real-time applications with delay and bandwidth constraints, such as multimedia, require such hard guarantee. An early example of bounded QoS is proposed in [63]. In this scheme a two-way lightpath reservation, along with a centralized scheduling technique, is proposed to provide bounded blocking probabilities. Other examples of absolute QoS schemes include early dropping and wavelength grouping schemes proposed in [83] and [84]. In the former, bursts of lower priority class are probabilistically dropped in order to guarantee the loss probability of higher priority class traffic. In the wavelength grouping scheme, the traffic is classified into different groups and a label is assigned to each group. A minimum number of wavelengths can be provisioned for each group. An edge-to-edge signaling and reservation scheme guaranteeing the edge-to-edge loss probability has been proposed in [77]. In this scheme, based on the available intermediate link states, the egress node uses a class allocation algorithm to assign each intermediate link a class supporting the related burst flows.

2.3.3 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. 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 changes. In threshold-based burst assembly approaches, on the other hand, 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. If the packet arrival rate is very high, a threshold-based burst assembly approach will generate bursts at non-periodic time intervals. More efficient assembly schemes can be achieved by combining the timer-based and threshold-based approaches

A major 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 still under investigation. If the threshold is too low, the bursts become very short and more bursts will be generated 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, 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.

In [56], [49], [57], [58], the authors consider a number of issues regarding burst assembly techniques. In [56], for example, 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, allowing early resource reservation in the OBS core; thereby, reducing the burst assembly delay. In [49], [57], [58], the authors study the impact of burst assembly on long range dependency of the input packetized traffic.

2.3.4 TCP Over OBS

A majority of data traffic in the Internet consists of TCP-based applications including Web (HTTP), Email (SMTP), peer-to-peer file sharing and Grid computing. Hence, OBS networks must be TCP-friendly in the sense that it must be able to handle the TCP-based applications without degrading TCP layer performance. A critical issue which can impact the TCP performance over OBS network is the random burst losses, which can be interpreted by the TCP layer as congestion in the network and hence may unnecessarily reduce the throughput, even at low loads. Another important issue which can have a significant impact on TCP performance is the effect of burst assembly in the OBS layer.

Recently, several works have evaluated TCP throughput over an OBS network. The impact of data burst assembly delay on TCP over an OBS layer has been investigated in [60]. Similarly, [87] examines the impact of data-burst lengths, burst-assembly times, and data burst drop rates. This study suggests that for low drop probabilities, increasing burst sizes results in higher throughput and increased delay. On the other hand, for high drop probabilities, there is no significant gain with increasing burst sizes. Other studies have proposed additional features for OBS networks, such as retransmission capability or burst acknowledgment, in order to improve the TCP throughput over OBS network. One way to achieve reducing the possibility of false congestion detection by the TCP layer is to retransmit data bursts at OBS layer, as proposed in [85]. Through simulations, it has been shown that the retransmission-based OBS can significantly improve the TCP throughput over OBS. In [82] a loss detection and error recovery mechanism by means of electrical buffering for OBS networks have been proposed and an analytical model to evaluate the TCP performance of an OBS network is presented.

Although TCP continues to be the dominant transport protocol and its support is essential for OBS networks, some researchers are rethinking some of the basic characteristics of IP protocols and attempting to develop novel transport protocols in order to take better advantage of OBS networks. This is motivated by OBS technology characteristics including high throughput, very low error rates, lack of buffering, and ability to handle bursts with variable lengths.

2.3.5 OBS Applications

Long before the development of OBS technology, burst switching concepts had been proposed as an extension to fast packet switching. Basic advantages of burst switching were reducing loop length and increasing data rate transmission [40]. In optical burst switching the concept of burst switching is extended to optical networks. The main motivation for such technology is to reduce (or eliminate) the need for optical buffering, as well as minimizing the network overhead. Consequently, OBS technology has been considered as the underlying network technology for various applications with large data requests and sensitive to path delay. One such application is distributed database. A distributed database is a collection of databases located at different geographic locations and connected through a network [59]. In these networks, large pieces of data from different locations must be aggregated for computation. Hence, minimizing the delay in data aggregation is a key issue in improving the overall system throughput. In such applications, optical burst switching technology can achieve efficient data assembly and path setup while reducing network overhead.

Another attractive area where OBS has been considered as an effective underlying technology is global Grid computing as a means of providing global distributed computing for applications with large bandwidth, storage, and computational requirements. A generic OBS-based architecture suitable to support Grid computing has been proposed in [64] and key issues such as signaling issues, anycast routing, and transforming jobs into individual data bursts are discussed. Such areas are subjects of many ongoing research activities.

As a final remark, we emphasize that the developed concepts and protocols for OBS networks are not limited to optical networks. Many of the basic aforementioned techniques and models developed for OBS network, can also be extended to sensor and satellite networks. For example, sensor networks can potentially benefit from similar assembly strategies and grooming techniques developed for OBS networks. In satellite communications, where the network is less delay sensitive and has limited number of satellite switch nodes, data packets transmitted between transponders can be aggregated into data bursts with out-of-band signaling. Such networks can be more flexible and efficient than traditional SS/TDMA-based (Satellite-Switched Time Division Multiple) networks in terms of offering wide-band capacity. Many of the contention resolution policies, scheduling algorithms, as well as QoS models, specifically developed for OBS networks can be potentially utilized for burst-based satellite networks.

2.4 Conclusion

In this chapter, we introduced synchronous and asynchronous OBS network architectures and described the basic design challenges in each case. We also identified the basic components of an OBS network and examined the architecture of edge nodes and core nodes. We provided a brief summary of various protocols dealing with critical issues in OBS networks. Furthermore, we discussed some of the practical applications proposed for OBS technology.

CHAPTER 3
A MULTI-LAYERED APPROACH TO OPTICAL BURST-SWITCHED NETWORKS

3.1 Introduction

In this chapter we represent the OBS network architecture in a layered manner as a set of protocols that can provide various services and exchange data with one other. A well-defined architecture with well-defined interfaces between the layers is essential for the practical implementation of OBS, as well as for the inter-operability of OBS with other networks. Furthermore, the layered hierarchy representation can provide a detailed insight into various implementation techniques, specifications, and functionalities of an OBS network. In addition to providing a layered view of OBS architecture, in this chapter we provide a brief summary of various protocols and algorithms addressing critical issues on OBS networks.

The reminder of this chapter is organized as follows. Section 3.2 discusses the layered architecture of IP-over-OBS. Section 3.3 describes each layer of the OBS layered architecture, separating them into a data plane and a control plane. Section 3.4 provides a layered view of an OBS network, illustrating an end-to-end transmission to show what role each layer plays in the data transmission. Finally, Section 3.5 concludes the chapter.

3.2 IP-over-OBS Layered Architecture

An important objective in the design of OBS networks is the large-scale support of different legacy services, as well as emerging services. In this section, without loss of generality, we will discuss the OBS network as it supports IP traffic; however, the OBS architecture described here is general enough such that it is capable of supporting most types of higher-layer traffic. Fig. 3.1 shows the layered hierarchy of an IP-based OBS network. We call this hierarchy the IP-over-OBS architecture. In this representation the IP layer treats the OBS

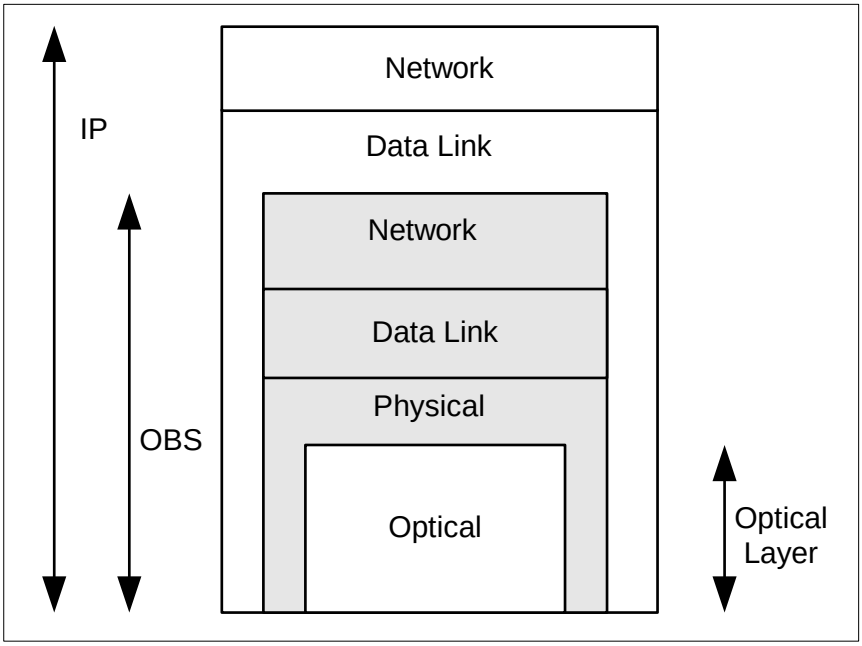


Figure 3.1. IP-over-OBS hierarchical layered architecture.

as its link layer, while the OBS operates on top of the optical (or DWDM) layer [88]. Thus, as a data transport system, the OBS network architecture implements the lower 3 layers, namely, physical, data link, and network layer. Fig. 3.2 shows our proposed OBS layered architecture, which follows the OSI reference model. In this representation we separate the control plane functionalities and protocols from those of the data plane. Such separation appears natural since the control information is transmitted out-of-band in OBS networks. Note that, in this model, we are ignoring the management plane, since the management plane communicates with all other layers and has no hierarchical relationship with them.

The control plane is responsible for transmitting control packets (CPs) while the data plane constructs and processes the data bursts (DBs). The CPs contain the information necessary for switching and routing DBs across the OBS network. The CPs are used for establishing the proper path prior to the arrival of the corresponding DBs, which arrive after some offset time. The CPs can also provide network management signaling.

Having two distinct planes suggests that each plane can operate independently of the other, using its own layers and protocols. Thus, it is conceivable to imagine that the DBs

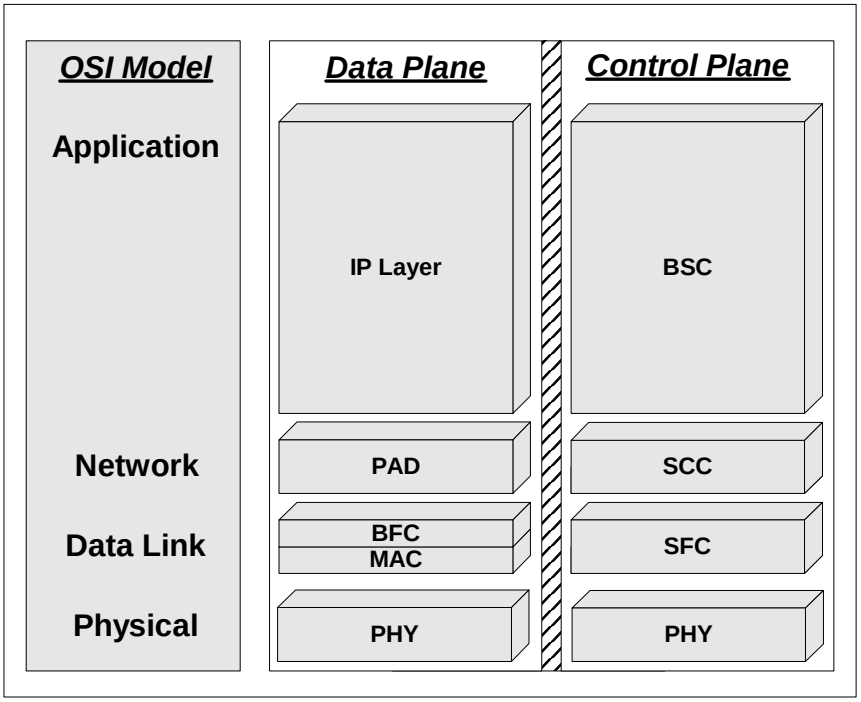


Figure 3.2. OBS layered architecture.

and CPs are encoded and routed on different transmission media.

In general, the OBS data plane architecture must take full advantage of DWDM technology and must support high capacity data transport links with no optical-to-electrical conversions. On the other hand, the design objective in the control plane is to make it flexible with low complexity. One way to achieve this goal is by processing CPs electronically. This approach offers high flexibility but limited processing capacity (a few tens of gigabits per second). Thus, simple encoding techniques and short frame lengths with minimum control overhead are required to allow fast and efficient CP processing. Transmitting CPs free of contention and in a highly reliable manner is also critical, since any error or loss of CPs results in higher data burst loss.

3.3 OBS Layered Architecture

In the following sections we describe basic functionalities of each layer in the data and control planes. We start with the data plane, which interconnects the OBS network with

other client networks. For clarity, we describe the layered architecture of each plane in an order consistent with packet flow.

3.3.1 Data plane layers

The data plane transports incoming packets from the edge source node to a single or multiple destination nodes. Line cards in the edge node provide an interface with packets arriving from various client networks. The line cards can perform error detection and error correction on incoming IP packet headers. Since in this section we only consider IP-based OBS networks, we assume that all packets entering and leaving the OBS network are IP packets, and that these packets maintain their original format and structure.

Packet Aggregation and De-aggregation (PAD) Layer

The PAD layer aggregates incoming IP packets of the same properties into data bursts. This layer also de-aggregates received data bursts into individual IP packets and assigns the packets to the proper outgoing link.

Transmitting IP packets at the ingress path of an OBS network requires determining individual packet properties and aggregating the packets together. Packet properties include packet Quality-of-Service (QoS) and its client destination address. After each incoming IP packet is decoded, its destination address must be translated to an OBS equivalent edge node address. Packets with similar properties are then aggregated to form the burst payload.

An important issue in OBS networks is data burst assembly. Burst assembly is the process of aggregating IP packets with the same destination into a burst at the edge node. The most common burst assembly techniques are timer-based and threshold-based. In timer-based burst assembly approaches [61], a burst is created and sent into the optical network at periodic time intervals; hence, the network may have variable length input bursts. In threshold-based burst assembly approaches [48], 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. A combination of timer and threshold-based approaches has been proposed in order to reduce the variation in the burst characteristic due to the variations of load [93]. In addition, a composite burst assembly approach [92] can be adopted in order to support QoS. 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.

In the egress path, the PAD disassembles data bursts into IP packets. Each packet's header must be processed for its destination address and the type of service it requires. The destination address is translated to identify which line card the IP packet must be sent to. Line cards, in turn, forward packets to the appropriate interfaced client network such as a LAN or WAN.

The PAD layer contains various flow control mechanisms and offers sequence verification of incoming data bursts. The flow control protocols can pace the rate at which DBs are placed on a link. If data burst deflection routing is allowed throughout the OBS network, then DB re-sequencing at the destination node may be required to ensure ordered delivery of IP packets.

Various protocols may be considered to perform address translations. Intelligent protocols can dynamically keep track of network configuration changes and support broadcasting transmissions. In addition, numerous admission control schemes have been proposed to address IP packet aggregation techniques. Packet aggregation may be based on a single or multiple packet properties such as destination, class, or flow. On the other hand, aggregation size remains as an important issue. For example, for low priority data bursts, a greater degree of aggregation results in greater loss of protection. While dealing with these concerns, such protocols must also reduce packet end-to-end delay and assure QoS without impacting the bandwidth efficiency.

A limited number of literatures have addressed data burst grooming in OBS. Data burst grooming can be an effective scheme to improve network performance when the packet arrival rate is low and data bursts (aggregation size) must maintain a minimum length due

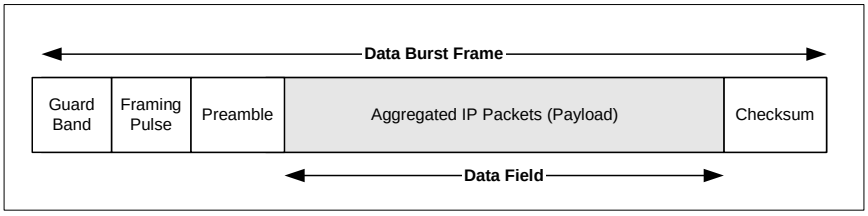


Figure 3.3. Data burst frame.

to core node's slow switching time. In [89] the authors consider data burst grooming at core nodes where several sub-bursts sharing a common path can be aggregated together in order to reduce switching overhead. The aggregated sub-bursts can be separated at a downstream node prior to reaching their final destinations. In [65] authors address the problem of data burst grooming at the edge node and focus on improving blocking probability and average end-to-end packet delay. They provide edge node architecture for enabling burst grooming and propose several data burst grooming heuristic algorithms.

Burst Framing Control (BFC) Layer

The function of the burst framing control layer is to receive the aggregated packets from the higher layer (PAD) and to encapsulate them into proper frame structures. This layer also decodes incoming data burst frames and extracts the data field. Fig. 3.3 represents a generic framing format of a data burst. When data burst frames have variable length and arrive at any random time, a framing pulse is necessary to indicate the beginning of each optical data burst frame. Framing pulses are typically isolated from the data-field by using a preamble to ensure data integrity.

Guard bands are normally a stream of fixed pulses used to separate consecutive frames. They are mandatory for reasons such as link length error, precision of clock distribution, and thermal effects. The checksum field may be required when data burst retransmission from the source to destination edge nodes is supported. In this case, edge nodes must be designed with considerable storage capacity. Use of the checksum may be considered especially when the medium does not offer the required transmission error rate.

The data field in the data burst frame can be further subdivided into fixed or variable sized segments. In this technique, which is referred as segmentation [92], the BFC inserts extra control information in each segment containing multiple IP packets.

Medium Access Control (MAC) Sublayer

The MAC sublayer in data plane includes the reservation and scheduling protocols, the offset time assignment protocols, the contention resolution schemes, and multicasting protocols. The MAC layer can also provide class differentiation in order to provide higher protection for DBs with QoS requirements. The actual signaling process by which a node requests the network to setup or release a connection is performed in the control plane.

An OBS network is inherently a point-to-point network in which adjacent nodes are interconnected to each other through direct physical links. However, asynchronous data bursts entering a core node from different links may need to access the same outgoing link. The MAC sublayer provides a way to control access to the outgoing links among these data bursts. In general, access control schemes proposed for OBS networks can be categorized as centralized or distributed.

In a centralized OBS network [62] a single node (called the request server) will be in charge of data burst transmission throughout the entire network. Clearly, this mode of operation makes medium access straightforward since the request server provides a single point of coordination that eliminates contention and packet loss. However, centralized scheme is very complex and considered to have low reliability and robustness.

In a distributed OBS network each node operates autonomously. This scheme suffers from lack of any centralized coordination. Consequently, the number of DBs entering a node and attempting to access the medium may exceed the number of available channels of the outgoing port. This is the primary source of contention in distributed OBS networks. Therefore, efficient and reliable algorithms in the MAC sublayer are required to simultaneously minimize contention as well as expected end-to-end delay of DBs.

Based on the type of service requested by an application, such as connection-less or

connection-oriented services, the OBS MAC needs to assign sufficient bandwidth and resources. Such assignments are obtained through the appropriate reservation protocols. Reservation protocols indicate the mechanisms in which a burst allocation starts and ends. Various out-of-band one-way reservation approaches have been proposed for OBS networks. The most widely considered signaling architecture for classical OBS are the Just-In-Time (JIT) reservation scheme and the Just-Enough-Time (JET) reservation scheme. Different variations of the JIT-based reservation schemes have been described in [76]. Although the JET-based OBS provides a more efficient use of bandwidth, its implementation requires higher complexity [45].

Various scheduling disciplines can be implemented in the MAC depending on the reservation protocols employed in the system. An OBS scheduling discipline determines the manner in which available outgoing data channels are found for DBs. Scheduling algorithms must be fast and efficient in order to lower the processing time and to minimize the data burst loss. Some common data channel scheduling algorithms for JET systems include first-fit unscheduled channel (FFUC) [100], latest available unscheduled channel (LAUC) or Horizon Scheduling, and latest available unscheduled channel with void filling (LAUC-VF) [39], [100]. More efficient void-filling mechanisms have been presented in [79] and [74]. In [81] a new signalling protocol is introduced which eliminates the generation of voids (or unscheduled blocks between data bursts) as data burst requests are scheduled. A complexity comparison between different scheduling mechanisms is provided in [98].

Scheduling protocols in the MAC layer should support class differentiation and provide a greater degree of protection and transmission reliability for high priority data bursts. We will discuss some of the proposed service differentiation schemes for OBS network in later sections.

In addition to addressing scheduling algorithms and contention resolution policies, another function of the MAC sublayer is offset assignment and maintenance between DBs and their CPs. The offset time can be variable or fixed. However, as the CPs are processed

and reconstructed at each hop, the offset times tends to be reduced. The core node must be able to account for such variations. The MAC protocols dealing with these issues are referred as offset control protocols.

A major concern in distributed medium access control scheme is high contentions. Packet transmissions in this scheme can only be statistically guaranteed. Many different techniques and algorithms have been introduced to improve OBS reliability and to reduce the DB drop ratio. A brief survey of different contention resolution schemes is provided in later sections.

The MAC sublayer can also support establishing multipoint multicast connections. In these schemes any edge node can transmit its DBs to multiple destination edge nodes. Efficient techniques for multicasting are becoming more popular and critical in the Internet for applications such as video-conferencing, video-on demand services, and content distribution. In general, similar contention resolution mechanisms implemented for unicast traffic can also be considered to support multicast traffic. For example, in [96], multicasting with deflection routing is considered. In [78], the basic focus is alleviating overheads due to control packets and guard bands associated with data bursts when transporting multicast IP traffic.

Physical (PHY) Layer

The physical layer of OBS is responsible for the actual transport of DBs and CPs from one node to another. It includes converting signals into appropriate electrical or optical format and uploading DBs into appropriate transmission frames. The physical layer also defines the actual physical interfaces between nodes in OBS. The PHY is divided into two sublayers:

- Data transport component,
- Medium dependent component.

We describe these sublayers briefly in the following paragraphs.

Data Transport Component (DTC): This is the medium independent sublayer of the physical layer. In the ingress direction it encodes data bits into specific pulse transmission called line codes (such as NRZ, AMI, HDB3, etc) and performs electrical/optical conversions. This sublayer also specifies transmission capacity.

Furthermore, DTC is responsible for implementing mechanisms to resolve synchronization issues between nodes including transmission techniques. Transmission techniques in OBS networks can be divided into two broad categories: slotted and unslotted. In synchronous slotted OBS networks CPs and DBs are only transmitted on their slot boundaries. In this transmission scheme, control and data channels are divided into time slots with fixed duration. Each control slot is further divided into several control slots with fixed duration. In an unslotted asynchronous network there is no need to delay a data burst and its BHP until the appropriate slot boundaries have arrived. In such networks each node has its own internal clock and DTC ensures sufficient inter-frame gap and defines the maximum allowable clock variation. The DTC also specifies the buffering requirement to alleviate any clock jitters among nodes.

Medium Dependent Component (MDC): This sublayer deals with the actual type of the medium used to transmit CPs and DBs including, coax cable, radio frequency, or optical fiber.[1] Selections of connectors, transmitters, receivers, etc., are considered as parts of the MDC sublayer. In an OBS network, as a special category of burst switching, the MDC is transparent to the photonic (WDM) sublayer, which provides lightpaths to the network. A lightpath is an end-to-end connection established across the optical network, and the lightpath uses a wavelength on each link in the path between the source and destination. Consequently, tasks such as optical amplification and wavelength conversion are defined in the MDC sublayer.

[1] Note that the concept of burst switching can be implemented on various mediums and it is not limited to optical burst switching.

3.3.2 Control plane layers

We now turn our attention from the data plane to the control plane. As we mentioned earlier, separation of planes in the OBS network architecture was inspired by the need to provide practical and reliable medium access protocols at high speeds. Due to current technological limitations in all-optical packet switching, it is not practical to implement MAC protocols in the data plane without interrupting data by optical-electrical converters. In OBS networks, implementing the MAC sublayer as the application layer of the control plane allows arbitration protocols to be performed in a domain (electrical) independent of data (optical).

The control plane in an IP-centric OBS network can be based on existing protocol standards. For example, similar to the Internet protocol, we can implement the Resource reSerVation Protocol (RSVP) and the Open Shortest Path First (OSPF) protocol in the control plane to provide signaling. Such standard protocols support a variety of control functionalities as well as multipoint multicasting. However, the major issue with implementing such protocol structures is their complexity and long processing time requirements. With this motivation, new protocols maybe considered to optimize control message processing and new signaling semantics. Key features of the new protocols must be flexibility and low complexity. In the following paragraphs we briefly describe general signaling semantics and basic functionalities of each layer in the control plane.

Burst Signaling Control (BSC) Layer

The BSC layer contains the data plane MACs' scheduling, contention resolution, and offset control protocols through its signaling protocols. Data burst properties including destination address, quality-of-service, etc., are passed to the BSC layer from the MAC sublayer. The BSC layer determines the type of the control packet to be transmitted to the next hop. Typical examples of the control packet types are burst header packets (BHP), burst cancelation packets (BCP), or network management packets (NMP). BHPs contain their associated data burst properties, BCPs can be used to cancel an existing reservation in downstream

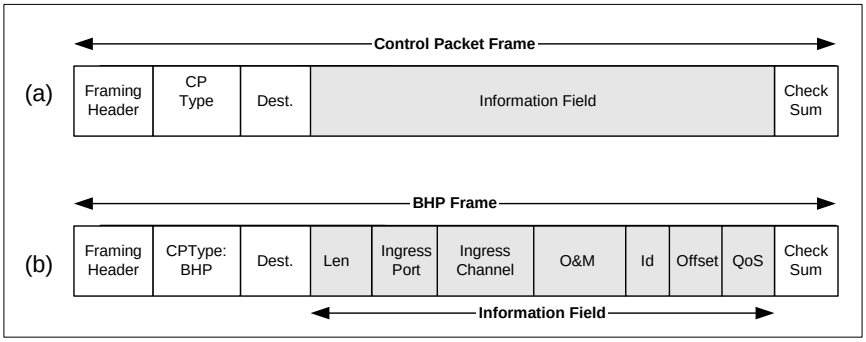


Figure 3.4. OBS framing structure of the control packet; (a) a generic control packet frame; (b) BHP frame.

nodes, and NMPs provide network status information. Other types of control packet can be considered to support multipoint multicasting connections.

Signaling Connection Control (SCC) Layer

The SCC layer includes the routing algorithms for control packets in order to establish the physical path for incoming data bursts. The actual data burst routing also takes place in this layer. Note that, in general, since the data and control planes can be implemented on separate mediums, it is possible that the physical routing paths for CPs and DBs are different. Various routing protocols can be considered for implementation in the SCC layer.

Signaling Frame Control (SFC) Layer

The main purpose of the SFC layer is to provide reliable transmission of control packets. The SFC layer can be considered as a pure data link protocol operating between adjacent nodes. The SFC layer receives bit streams containing the control packet type and its associated data burst properties, and it constructs CP frames by attaching overhead bits. Many popular framing mechanisms such as High-Level Data Link Control (HDLC) may be considered for the data link protocol. However, the protocol complexity and cost are critical as interface speed increases. Fig. 3.4(a) shows a generic framing format of a control packet frame.

Typically, control packets in OBS networks are continuous fixed-size packets, which

are processed electronically. Therefore, there is no need for attaching a framing pulse and for the use of a preamble. However, each CP must still contain its own framing header.

To guarantee fast processing of control packets at each node, CPs must contain limited information, yet, it is crucial to protect control packets from errors on each link. Transmission errors in control packets can result in bits being changed in the information field. Incorrect bits will be misinterpreted in the downstream core node and result in, for example, dropping high priority bursts, incorrect switch fabric setup, or even burst misrouting. To protect the CP from error, a cyclic redundancy check (CRC) can be implemented in the checksum field. CRC codes can provide a large selection of error correcting capacity [86]. Each CP must also have a destination field indicating its destination node. Furthermore, all CPs must have a type indication specified in the CP-type field. Different CP types were described in the Burst Signaling Control section. Contents of the information field vary depending on the CP type.

If a control packet is associated with an incoming DB, it is referred to as a BHP. A typical BHP frame is shown in Fig. 3.4(b). Note that the BHP information field is divided into several fields including length, ingress port, and ingress channel, which refer to DB's duration, its edge node source, and the wavelength on which it is expected to arrive, respectively. The id field can be useful for checking data burst sequencing when deflection routing is allowed. The QoS and offset fields indicate the incoming data burst priority level and the offset time between a BHP-type control packet and its associated data burst, respectively. The O&M field contains network management related signaling information, such as loop-back requests, protection switching, or link failure notification.

Physical (PHY) Layer

The physical layer in the control plane performs similar functionalities to the data plane's physical layer but it may have different characteristics. One such difference is the transmission rate. Control packets can be transmitted at lower rate than data bursts in order to achieve practical packet processing. CPs' transmission rate and offset time are generally

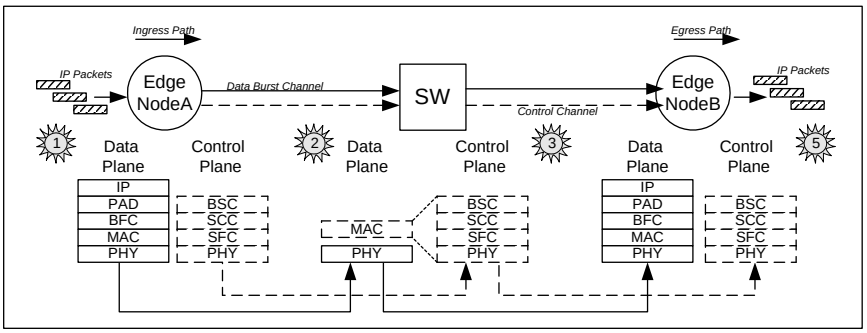


Figure 3.5. Transport stages in an OBS netowrk.

designed for optimum performance in terms of end-to-end delay and bandwidth efficiency.

In addition, as in the data plane, the PHY layer addresses synchronization issues and determines transmission techniques such as slotted or unslotted transmission. In general it is convenient to implement the same transmission technique in both data and control planes.

3.4 An Example Multi-layer Architecture in an OBS Network

In this section we attempt to illustrate our proposed OBS layered hierarchy by means of an example. Fig. 3.5 shows various transport stages in a simple OBS network configuration where DBs and their control packets are transmitted on the same link but on different wavelengths and are separated in time by an offset. We assume that each link contains a single control channel, the core node is bufferless, and data bursts are IP-based. For convenience, Appendix A (at the end of this chapter) provides a summary of all layers in data and control layer.

Stage 1 - IP Interface: The edge node's line cards receive IP packets from various client networks. They process packet headers and extract the type of service and the destination client address. The aggregation protocol in the PAD layer translates the destination client address into an OBS edge node address and determines the next hop. The protocol also classifies packets based on their destination, type of service, or both.

Stage 2 - Burst Aggregation: The PAD layer aggregates IP packets of the same class into bursts. An admission control protocol examines bursts and verifies whether or not they are ready for transmission. Upon completion of the aggregation process, bursts will be passed on to the BFC layer. The BFC encapsulates the aggregated IP packets into data burst frames and places them into the DB buffers. DBs are stored until the MAC sublayer assigns the DBs to a transmission time slot. The MAC sublayer sends the data burst properties to the BSC layer in the control plane. The BSC, in turn, constructs a control packet of type BHP and passes it to the SCC layer. The SCC layer assigns a routing path based on the DB destination address.

The BHP is handed to the SFC layer to be placed in the proper framing format. The BHP frame is stored in the BHP buffer and waits for the SFC to search for the next available time slot on the control channel. Once a potential BHP time slot is found, the SFC is required to verify the bandwidth availability on data channels based on the offset value (this is only required if the offset is not fixed). Assuming that the scheduling was successful, the BHP frame is passed on to the physical layer and transmitted on the link. After an offset time, the DB is sent to the physical layer on the data plane and transmitted on the pre-assigned channel.

Stage 3 - Burst Switching: The core node receives the BHP and processes it electronically. The SFC verifies the BHP's checksum and, assuming the checksum matches the calculated value, extracts information fields from the frame. The SCC identifies the requested connection in the core node and its duration. This reservation request is sent to the BSC layer and verifies available bandwidth and decides to either make the reservation or to discard it. If the reservation was successful then the reservation table in the switch fabric control unit will be updated. The expected DB arrives after the offset time and it cuts-through the pre-established path in the optical switch. Thus, the DB only goes through the physical layer of the data plane without any O/E interruptions.

Similar steps as described in the burst aggregation stage take place in order to transmit the control packet on the outgoing control channel. The BSC creates a new control packet

of type BHP and the SCC assigns the next hop's outgoing port. The CP is encapsulated into a frame and transmitted on the assigned time slot. On the other hand, data bursts only go through amplification and wavelength conversion, which may be required for compensating power loss and accessing the outgoing port, respectively. Note that, as shown in Fig. 3.5figure 7, by processing the CPs in the control plane, we can essentially implement the data plane MAC sublayer without having to process DBs. The loss of power occurs as optical signals are decoupled and traverse through the optical switch fabric.

Stage 4 - Burst Disassembly and IP Forwarding: At the destination node all CPs and DBs will be terminated and processed electronically. The CPs are verified for errors in the SFC layer and upon detecting any errors, their associated DBs will be discarded (consequently, the destination edge node may request retransmission). Similarly, data channels are de-multiplexed and DBs are verified for transmission errors and de-framed in the BFC layer. The data burst payload is passed to the PAD layer and decomposed into individual IP packets. This layer can also check for DB order and decide what action to take (such as buffering or discarding) with out-of-sequence DBs.

Disassembled IP packets are processed to identify the client network to which they should be forwarded. The PAD layer needs to translate these addresses and determine the destination line card. The line card, in turn, forwards the packets to the proper client networks.

3.5 Conclusion

Optical burst switching has been proposed as a practical approach for supporting the next-generation high-speed high-capacity Internet. A layered architectural representation of the OBS network can be used as a baseline for understanding protocol requirements as well as their future development and design. In this chapter we provided an organized decomposition of the different layers for supporting OBS networks. Detailed descriptions of each layer along with their functionalities and related protocols were presented. To furnish a better understanding of the proposed layered architecture, an illustrative example

of an end-to-end data transmission was provided. The proposed layered architecture can be used as a baseline for future development and design of protocols and interfacing functions over optical burst-switched networks.

3.6 Appendix A: Summary of different sub-layer protocols in data and control planes.

Plane	Description
Data Plane	**IP Layer :** Receiving and transmitting IP packets **Packet Aggregation and De-aggregation (PAD) :** Classifies Packets based on their QoS requirements and destination Aggregates IP packets into super-packets Translates IP address to OBS nodes Disassembles packets into individual packets Verifies DB sequence Controls DB transmission flow **Burst Framing Control (BFC):** Encapsulates the super-packets into burst frames **Medium Access Control (MAC):** Provides methods to access outgoing ports Assigns sufficient BW and resources for each request Handles contention resolution schemes through the control plane Supports multicasting Determines the offset time value **Physical Layer (PHY):** Handles synchronization schemes and clock recovery issues Determines transmission rate and capacity Specifies transmission technique (slotted, unslotted) Signal conversions such as Optical Electronic
Control Plane	**Burst Signaling Control (BSC):** Responsible for implementing scheduling and contention resolution protocols Generated control packets Implements the signaling scheme **Signaling Connection Control (SCC):** Determines routing and forwarding algorithms **Signaling Frame Control (SFC):** Implements the hop-to-hop data link protocols Constructs the control packet frame **Physical Layer (PHY):** Handles synchronization schemes and clock recovery issues Determines transmission rate and capacity Specifies transmission technique (slotted, unslotted) Signal conversions such as Optical Electronic

CHAPTER 4

ANALYSIS AND IMPLEMENTATION OF LOOK-AHEAD WINDOW CONTENTION RESOLUTION WITH QOS SUPPORT IN OPTICAL BURST-SWITCHED NETWORKS

4.1 Introduction

Recently, considerable attention has been given to address and study various important issues in OBS networks. For example, many articles have focused on signaling and scheduling mechanisms for reserving and releasing resources in OBS. First-Fit, Horizon, Latest Available Unscheduled Channel (LAUC), and Latest Available Unscheduled Channel with Void Filling (LAUC-VF) are among the proposed scheduling algorithms [39], [100]. In both LAUC and LAUC-VF scheduling algorithms, a burst chooses the unused channel that becomes available at the latest time. When void filling (VF) is allowed, gaps between two scheduled data bursts can also be utilized. In these schemes the data burst reservation time starts at the beginning of the actual burst arrival and lasts until the end of the burst. Some studies have been dedicated to OBS architecture issues, including the signaling protocols and scheduler architecture [100], [68]. Others have proposed various ways to implement Multi-Protocol Label Switching (MPLS) and TCP/IP over OBS [115], [133], [134].

A major concern in OBS networks is high contention and burst loss. Typically, there are two main sources of burst loss: contention on the outgoing data channels and contention on the outgoing control channel. In this chapter we focus on output data channel contention, which occurs when the total number of data bursts going to the same output port at a given time is larger than the available channels on that port. Contention is aggravated when the traffic becomes bursty and when the data burst duration varies and becomes longer.

Contention and loss may be reduced by implementing contention resolution policies. There are different types of contention resolution techniques, such as time deflection (using buffering) [101], [62], space deflection (using deflection routing) [102], and wavelength

conversion (using wavelength converters) [103]. When a contention cannot be resolved by any one of these techniques, one or more bursts must be dropped. The policy for selecting which bursts to drop is referred to as the *dropping policy*.

A dropping algorithm may be utilized in conjunction with a scheduling algorithm to protect high priority bursts while reducing the overall burst loss rate. Thus, the dropping algorithm is invoked only when no available unscheduled channel can be found for a BHP request.

Two well-defined dropping algorithms have been proposed. One is based on dropping the latest arrival and the other is based on dropping only the portions of the burst involved in contention. In this chapter we elaborate on the performance of each of the above schemes as well as their QoS supporting capacity. We also introduce two new algorithms capable of handling service differentiation: look-ahead and shortest drop contention resolution. The first contribution of this chapter is to provide an efficient algorithm in order to resolve contention while minimizing burst loss.

Any contention resolution algorithm must also be practical for high-speed hardware implementation in terms of processing time, scalability, and cost. The algorithms processing time is directly proportional to its complexity and directly impacts the end-to-end packet delay. By reducing the processing time, the optical buffering requirements can also be minimized. The contention resolution algorithm implemented in the scheduler unit of the core switch must be scalable in order to sustain system growth as new ingress and egress ports and channels are added. The primary cost of the algorithm's, and hence scheduler's, implementation is its memory requirements, which highly depends on the complexity of reservation algorithms. Many efforts have been made to propose practical design solutions to OBS signaling protocols [72] and channel scheduling algorithms [105].

Hence, the second contribution of this chapter is to present a practical hardware architecture of the scheduler block capable of implementing the contention resolution algorithm and suitable for the core node. We focus on three objectives: first, the design must be generic such that it can operate with any contention resolution algorithms; second, the de-

sign must be such that it can fully be implemented in hardware and involves no software; finally, the design must be realizable on an available off-the-shelf reconfigurable hardware device, i.e. a field programmable gate array (FPGA) operating at hundreds of MHz. With such objectives in mind, we implemented the look-ahead contention resolution algorithm in a single high-density FPGA device and evaluated the design in terms of cost, processing speed, and scalability.

In any contention resolution algorithm, fairness is considered to be an important issue. A fair contention resolution algorithm does not discriminate between different traffic sources and treats bursts with different lengths similarly, while minimizing the overall packet loss. In this study, we do not address the issue of fairness and defer it to future investigation.

The rest of this chapter is organized as follows. In Section 4.2, we first briefly describe the system configuration under consideration as well as the core node architecture. Then, we provide detailed descriptions pertaining to several proposed contention resolution algorithms. In Section 4.3, the efficiency of our proposed contention resolution algorithm is evaluated analytically. The performance results through computer simulation for each of the introduced algorithms are provided in Section 4.4. In Section 4.6, we describe a practical general purpose hardware architecture design for the scheduler and we examine the performance of our algorithm within this architecture under emulated traffic. Finally, Section 4.7 concludes the chapter.

4.2 Description of Dropping Algorithms

In this section we describe the network under study, and discuss details of different dropping algorithms.

4.2.1 Network assumptions

The network under discussion in this section consists of a number of edge nodes connected to a core optical network with no buffering capacity. We assume that each link has a single

control channel and multiple data channels. A detailed architecture of the edge nodes is provided in [100] and [104].

The data burst transmission scheme can be either slotted or unslotted [68]. In this section, we assume slotted transmission in which data bursts and their corresponding BHPs are transmitted only on the slot boundaries. Consequently, the offset time and the duration of a data burst will be interpreted in units of slots. Furthermore, we assume that incoming data bursts have different service types with different QoS requirements.

Fig. 4.1 shows the generic core switch node architecture in an OBS network. In this hybrid architecture, the core switch is fully transparent to optical data bursts, while the control channels are converted into electrical signals. Fiber delay line (FDL) blocks can be used to compensate for BHP processing time delay in which data bursts are deliberately delayed in order to maintain the necessary offset times. Various switch fabric architectures have been discussed in different sections, including [52], [53] and [54].

In the core switch node architecture, shown in Fig. 4.1, incoming BHPs on control channels are processed and regenerated in the *BHP processor-regenerator block* (BPRB). In the BPRB the BHPs are first converted into electrical signals and then sent to a *control packet processor* (CPP), where they are processed and scheduled if proper resources are available. If a BHP request was successfully reserved, the switch fabric setup needs to be updated as the corresponding data burst arrives and leaves the switch. Furthermore, each accepted BHP must be regenerated with the updated information and transmitted to the downstream core node. The control packet processor is considered as the main part of the core node's BPRB and contains the data burst contention resolution algorithm.

4.2.2 Latest Arrival Drop Policy (LDP)

The simplest dropping policy is the latest-arrival drop policy (LDP). In LDP, the algorithm searches for an available unscheduled channel (as in LAUC-VF), and if no such channel is found, the latest incoming data burst will be discarded. Although the processing speed of BHPs in the LDP scheme is attractive, the main disadvantage of this technique is that it has

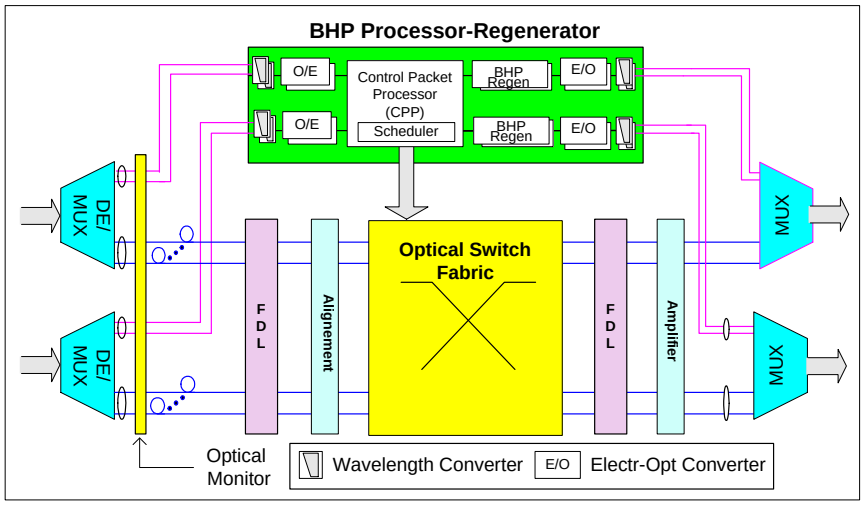


Figure 4.1. Typical architecture of the OBS core switch node with the header packet processor.

relatively poor performance with respect to data loss when no buffers are utilized.

Inherently, LDP is not capable of differentiating packets with different priority types. A novel scheme proposed by [106] suggests that giving extra offset time to high priority data bursts can increase the likelihood of their early reservations. This approach is known as offset-time-based QoS. The extra offset time must be large enough to ensure that the blocking of high-priority bursts by any lower-priority burst is minimized. Therefore, offset-time-based QoS is a tradeoff between guaranteeing lower loss for high priority data bursts and increasing their end-to-end delay.

4.2.3 Look-ahead Window Contention Resolution (LCR)

The look-ahead contention resolution algorithm takes advantage of the separation between the data bursts and the burst header packets. By receiving BHPs one offset time (Δ) prior to their corresponding data bursts, it is possible to construct a look-ahead window (LaW) with a size of W time units (slots). Such a collective view of multiple BHPs results in more efficient decisions with regard to which incoming bursts should be discarded or reserved. On the other hand, at each hop, the BHPs must be stored for duration of W time units before they are retransmitted (thus requiring $\Delta \geq W$). Fiber delay lines (FDLs) can be used on each hop to delay data bursts by W time units to maintain the original offset time.

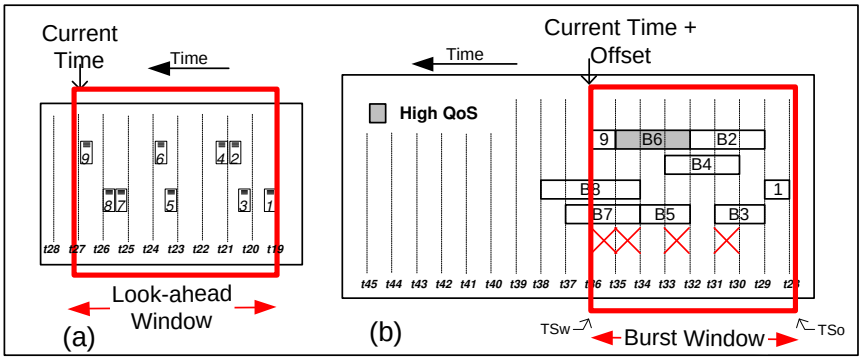


Figure 4.2. Look-ahead and burst windows for all bursts going to the same switch output port with 2 channels; Δ=9, W=8, L_{max}=4; $\times$ indicates contending regions.

Fig. 4.2(a) shows an example of the received BHPs for data bursts that are destined for the same switch output port with two available channels. Without loss of generality, we assume slotted transmission with window size W=2 $\cdot$ L_{max} time slots (t_{19} through t_{27}), where L_{max} is the maximum data burst duration in units of time slots. Using the received burst header information, a burst window can be constructed, as shown in Fig. 4.2(b), to describe the state of the switch one offset time later ($t_{19} + \Delta$ through $t_{27} + \Delta$). Once the burst arrival times within the burst window are determined, the LCR algorithm is applied to the entire burst window range. We define TS_o and TS_w as the starting and ending slots of each burst window, respectively. The algorithm finds the contending slots and then identifies which bursts should be discarded and which should be scheduled. The final dropping decisions are only applied to bursts whose starting times are the same as the start of the burst window at TS_o (e.g., burst B_1 in Fig. 4.2(b)). The LaW and burst window are advanced one slot at a time.

The look-ahead contention resolution algorithm can be divided into three basic steps: (a) collecting all BHPs destined to the same output port and creating a look-ahead window of size W; (b) determining the contention regions (slots), CR, in each corresponding burst window; (c) applying a heuristic algorithm to decide which of the contending data bursts within the burst window must be discarded.

Once the LaW is constructed and the arrival and departure times of the incoming bursts are determined, the contention resolution problem can be reduced to the following: if the

number of bursts directed to the same outgoing port on the switch exceeds its available channels, w, how can we resolve the contention while minimizing the burst blocking probability?

The contention problem can be solved by creating an auxiliary directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ representing all the bursts in the LaW. In this representation, the finite set $\mathbf{V}$ of vertices (nodes) identifies the starting and ending times of the bursts. That is

$$\mathbf{V} = \{(\mathbf{t_s}(\mathbf{1}), \mathbf{t_s}(\mathbf{2}), ..., \mathbf{t_s}(|\mathbf{V}|), \mathbf{t_e}(\mathbf{1}), \mathbf{t_e}(\mathbf{2}), ..., \mathbf{t_e}(|\mathbf{V}|)\}, \tag{4.1}$$

where $t_s(i)$ and $t_e(i)$ are the starting and ending times of data burst B_i, respectively, and $|V|$ is the number of bursts in the LaW. The finite set $\mathbf{E}$ of directed edges includes one directed edge for each burst. An edge exists between vertices $t_s(i)$ and $t_e(i)$ to represent burst B_i. The weight of each edge is equivalent to the duration of its corresponding data burst, $L(B_i)$. Furthermore, we define a set of contention regions, $\mathbf{CR} = \{\mathbf{CR_1}, \mathbf{CR_2}, ...\mathbf{CR_u}\}$ within the burst window, where a contention region, CR_i, is defined as a continuous set of slots in which there is contention. By finding the least-cost path from the beginning of the first contention region to the end of the last contention region (CR_1 through CR_u), we can find a set of data bursts that, if dropped, can resolve the contention while minimizing data loss.

In order to implement the shortest-path algorithm, we need to alter the original digraph $\mathbf{G}$ such that it is *connected*. Therefore, we introduce a set of zero-weight directed edges ($Z_{k+1,k}$), called zero paths, between adjacent nodes $k+1$ and k. Zero paths are added between unconnected adjacent nodes where *no* contention exists. Such cases typically occur when there is a short halt in data burst transmissions. In case an edge (a burst) already exists between adjacent nodes k and $k+1$, no $Z_{k,k+1}$ will be required.

The adjacent nodes within contention regions may also be disconnected. This is because overlapping data bursts can end on different time slots. Thus, in order to ensure graph connectivity within contention regions, we can add directed return paths, $N_{k+1,k}$, between adjacent nodes $k+1$ and k. We now describe the scheme in which the weight of return paths can be determined. Let us define the contention degree, d_{TS}, as the number of unsuccessful

bursts contending for an outgoing port in time slot TS. The shortest-path solution should remove as many as d_{TS} overlapping data bursts on each time slot TS in the window. This can be emphasized by having i negative-weight directed return paths between adjacent nodes $k+1$ and k ($N^{(i)}_{k+1,k}$, with $i = 0, 1, ..., d_{TS} - 1$). Having $i = 0$ implies that only one contending data burst must be removed. Thus, in this case, we can assume that the weight of the return path is zero ($|N^{i}_{k+1,k}| = |N_{k+1,k}| = 0$). On the other hand, when $i \geq 1$ the weight of each return path can be defined as

$$|N^{i}_{k+1,k}| = (-1) \cdot |E^{j-i-1}_{k,m}|, \tag{4.2}$$

where $|E^{j}_{k,m}|$ is the weight of the outgoing edge, E^{j}, from node k to another arbitrary node m such that

$$|E^{j+1}_{k}| > |E^{j}_{k}|. \tag{4.3}$$

The resulting connected digraph, including the zero and return directed paths, can be represented by $\acute{\mathbf{G}} = (\mathbf{V}, \acute{\mathbf{E}})$. The shortest-path algorithm can now be solved for $\acute{\mathbf{G}}$. Consequently, the solution can be obtained by implementing the Bellman-Ford algorithm which has a complexity of $\Theta(|\mathbf{V}|.|\acute{\mathbf{E}}|)$. Other variants of the Bellman-Ford algorithm can also be considered. In either case, special care must be taken to ensure that no negative cycles occur. It is important to emphasize that a single iteration of the shortest-path solution in the LCR may not remove all the contending data bursts. As a result, the LCR algorithm may have to be executed in multiple iterations until all contending slots are resolved.

Once the LCR algorithm is completed and all contention regions are resolved, a set of data bursts, $\mathbf{P} = \{\mathbf{B_x}, \mathbf{B_y}, ...\}$, is obtained for possible discard. However, only the bursts with starting time equal to current time can be permanently dropped.

The LCR algorithm can be readily modified to support service differentiation. Let us assume that the class type for a data burst B^{c}_{i} is defined by c, with c_{max} being the lowest priority level. The starting and ending slots of burst B_i are denoted by $t_s(i)$ and $t_e(i)$, respectively. In this case, the weight of the edge connecting node pairs $t_s(i)$ and $t_e(i)$ in

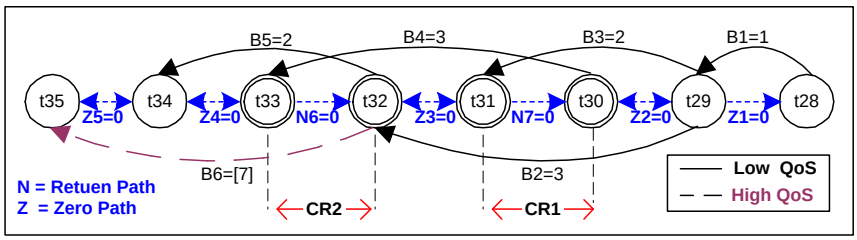


Figure 4.3. Directed graph, $\acute{\mathbf{G}}$= ($\mathbf{V}$, $\acute{\mathbf{E}}$), partial representation of example shown in Figure 1; for simplicity bursts with $ts(i)$ beyond t_{35} are not shown; L_{max}=4; B_6 is assumed to have high priority (c=1) and c_{max} = 2. Note that B_7, B_8, and B_9 are not shown due to lack of space.

graph $\acute{\mathbf{G}} = (\mathbf{V}, \acute{\mathbf{E}})$ can be a function of the duration and the priority level of burst B_i. That is

$$E_{t_s(i),t_e(i)} = [c_{max} - c] \cdot L_{max} + L(B_i^c). \tag{4.4}$$

Note that when all bursts have the same class priority, the edge weights become equivalent to burst durations. Several advantages can be attributed to the QoS-enabled LCR. For instance, it can support unlimited number of classes of service without extra offset time. The LCR mechanism can offer absolute as well as proportional class isolation. In absolute class isolation the possibility of a high-priority burst being blocked by any lower priority burst is eliminated. On the other hand, in proportional class isolation the dropping criteria will be based on the relative length and priority level of data bursts. In such a scheme, it is possible that between a short duration high priority burst and a long duration low priority burst, the one with higher priority will be discarded. Clearly, in terms of complexity, minimal additional steps are required to enable service differentiation in LCR.

At this point, we demonstrate the LCR approach by referring to the example shown in Fig. 4.2. We start by creating a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ The set of bursts within the look-ahead window is represented by $\mathbf{B} = \{\mathbf{B_1}, \mathbf{B_2}, ...\mathbf{B}_{|\mathbf{V}|}\}$, with $|V| = 9$. Thus, there will be 9 edges with 11 distinct nodes in $\mathbf{G}$, where $\mathbf{V} = \{\mathbf{t_{28}}, \mathbf{t_{29}}, \mathbf{t_{31}}, ..., \mathbf{t_{38}}\}$ and $\mathbf{E} = \{(\mathbf{t_{28}}, \mathbf{t_{29}}), (\mathbf{t_{29}}, \mathbf{t_{31}}), (\mathbf{t_{29}}, \mathbf{t_{32}}), ...\}$. Moreover, $\mathbf{CR} = \{\mathbf{CR_1}, \mathbf{CR_2}\}$, where $CR_1 = (t_{30}, t_{31})$ and $CR_2 = (t_{32}, t_{33})$. Each edge $(t_s(i), t_e(i))$ is assigned a weight representing

the burst B_i duration and its priority level. Using Eqn. (4.4), assuming $c_{max} = 2$ and $L_{max} = 4$, the weight of the edge (t_{32}, t_{35}) representing B_6 will be 7.

Fig. 4.3 depicts the modified digraph $\acute{\mathbf{G}}$ after adding the zero and return paths. Note that directed return paths of $N^0_{t_{31},t_{30}}$ and $N^0_{t_{33},t_{32}}$ are equivalent to zero-weight paths of N_6 and N_7, respectively. This is because the degree of contention in these time slots is one. Zero paths Z_1, Z_2, Z_3, Z_4, Z_5 are assigned because they are within non-contention regions, as described above. Note that $Z_{t_{28},t_{29}}$ has been replaced by the edge representing B_1. Solving the shortest-path problem, we find $\mathbf{I} = \{\mathbf{B_4}, \mathbf{B_7}\}$. This indicates that by discarding B4 and B_7 all contentions can be eliminated. However, since none of these data bursts arrive at the starting slot of the window ($TS_o = t_{28}$) no burst will actually be dropped until the window reaches the start of either burst.

An important issue pertaining the performance of the LCR is its fairness. Fairness indicates whether LCR tends to treat all bursts similarly or favor bursts with certain characteristics. For example, if shorter bursts tend to be dropped more frequently, then sources with lower packet rates, generating short bursts, will be discriminated against. In this study, we do not examine LCR's fairness. However, we believe, the LCR is more likely to favor longer bursts and drop short bursts.

4.2.4 Shortest Burst Drop Policy (SDP)

In order to reduce the end-to-end data burst delay in LCR, different variations of this algorithm can be considered. The tradeoff, of course, will be performance degradation. For example, if we reduce the window size to $W < 2 \cdot L_{max}$, the incoming data bursts can experience shorter per-hop delay, but the advance viewing capacity of the window will be decreased. Hence, the window size can be minimized to a single slot. In this scheme, each incoming burst slot will be checked, and upon detecting contention, the lower priority burst with the shortest duration and latest arrival time will preferentially be dropped. This allows BHPs to be processed and transmitted soon after they are received. We call this scheme the shortest drop policy (SDP).

One drawback of such a policy is its potential over-reserving of resources, since some earlier reservations may be eliminated later. In other words, the reservation for BHP_i can potentially be canceled within the next Δ_i slots before the associated data burst arrives. A simple way to reduce such over-reservations is to use *cancelation* packets to release down stream resources as soon as a burst is dropped.

In terms of supporting class differentiation, SDP can support unlimited number of priority levels and requires no extra offset assignments for bursts with higher service requirements. It also guarantees complete class isolation. In addition, SDP offers proportional differentiation, as described before.

4.2.5 Segmentation Drop Policy (SEG)

The basic assumption in this scheme is that each transmitted data burst consists of individual independent segments such as slots. Therefore, if contention occurs, only the segments of the lower priority burst involved in the contention will be removed. Details of this mechanism, known as *segmentation*, along with its variations are described in [92]. Although the QoS-enabled segmentation algorithm appears to be straightforward, the hardware implementation suffers from multiple issues. For example, due to segments being dropped, packets can arrive out of order. Furthermore, each segment must have its own framing headers and sequence number. This requires additional hardware complexity in terms of both burst assembly and disassembly process.

4.3 Algorithm Analysis

4.3.1 LCR NP-Completeness

In this section we show that the general problem of burst contention resolution in LaW, denoted by CRLaW, is NP-complete. We can redefine the graph model for this problem as a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is the set of all bursts in LaW, and any edge $e \in \mathbf{E}$ represents a *time overlap* between two data bursts in the LaW. In such a non-directed loopless graph model, the weight of each vertex, denoted by $\mathbf{W}(\mathbf{B_i})$ where $B_i \in \mathbf{V}$, will be equivalent

to the value assigned to each burst, including its length and the value of assigned priority level, as indicated by Eqn. (4.4).

Fig. 4.4(a) shows such a graph for the example in Fig. 4.2(b). In the context of such a graph model, we define the CRLaW problem as follows.

Definition 1: (CRLaW problem) Given a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ representing bursts in the LaW and their overlaps, determine the first largest w sets of non-overlapping bursts, $\bar{\mathbf{I}} = \{\mathbf{I_1}, \mathbf{I_2}, ...\mathbf{I_m}\}$, $\mathbf{I} \subseteq \mathbf{V}$, and $m \geq \lambda$.

Hence, we are looking for as many as w maximum-size sets of non-overlapping bursts, each of which can be scheduled on the same wavelength. We define a non-overlapping set, $I_j \subseteq \mathbf{V}$, as an *independent set* such that no two vertices in I_j are joined by an edge in $\mathbf{E}$. The value of w is equivalent to the number of wavelength channels on each egress link.

In order to prove the NP-completeness of the CRLaW problem, we first consider a special case in which all bursts have the same length and priority level.

Theorem 1: The CRLaW problem, when all bursts have the same weight, is NP-complete.

Proof: We need to reduce a known NP-complete problem to CRLaW problem. The known problem in this case is the Clique problem [107]. The Clique problem is stated as follows. Given an undirected graph $\overline{\mathbf{G}} = (\overline{\mathbf{V}}, \overline{\mathbf{E}})$ with $\overline{\mathbf{V}} = \{\mathbf{B_1}, \mathbf{B_2}, ...\mathbf{B_n}\}$ and a collection of $\bar{\mathbf{I}} = \{\bar{\mathbf{I}}_\mathbf{1}, \bar{\mathbf{I}}_\mathbf{2}, ...\bar{\mathbf{I}}_\mathbf{m}\}$ such that each element in $\bar{\mathbf{I}}$ contains a subset of $\overline{\mathbf{V}}$, is there a maximum size subset $\overline{I_j} \subseteq \overline{\mathbf{V}}$ such that all members of $\overline{I_j}$ are connected together? This is equivalent to asking if we can obtain a subgraph $\overline{I_j}$, which is a complete graph with maximum size such that $\overline{I}_j \cup \{B_i\}$ is no longer a complete graph for any B_i that is not in $\overline{I}_j$.

We now construct a graph $\mathbf{G}$ for an arbitrary instance of the Clique problem such that $\mathbf{G}$ contains a maximum independent set of size $\geq K$ if and only if $\overline{I}$ is a Clique in $\overline{\mathbf{V}}$. The following steps can be taken to construct such a graph:

- For all elements $B_i \in \overline{\mathbf{V}}$ and $e \in \mathbf{E}$, find subgraphs which are complete.

- Take the complementary graph of $\overline{\mathbf{G}}$ such that, for the same vertices, the edges of the new graph will be complementary to $\overline{\mathbf{E}}$.

We demonstrate this through the example shown in Fig. 4.4(b). Note that maximum size Cliques in Fig. 4.4(b) will be $\bar{\mathbf{I}} = \{\bar{\mathbf{I}}_{\mathbf{1}}, \bar{\mathbf{I}}_{\mathbf{2}}, ...\bar{\mathbf{I}}_{\mathbf{7}}\}$ where $\overline{I_1} = \{B_2, B_6, B_9\}$, $\overline{I_2} = \{B_3, B_5, B_8\}$, $\overline{I_3} = \{B_3, B_5, B_7\}$, etc.

Comparing Fig. 4.4(a) and (b), it is clear that a set of nodes is an independent set in $\mathbf{G}$ if and only if it is Clique in graph $\overline{\mathbf{G}}$. Therefore, finding maximum Clique in $\overline{\mathbf{G}}$ is equivalent to finding the maximum independent set in the original graph, $\mathbf{G}$. Consequently, the CRLaW problem is proven to be NP-complete. $\diamondsuit$

The weighted version of the CRLaW problem is the general case when bursts can have different lengths and priority levels. Therefore, the general weighted case of the CRLaW problem will have *at least* the same computational complexity as its unweighed counterpart. Consequently, we can state the following lemma.

Lemma 1: The weighted CRLaW problem is NP-complete.

Let us now consider the case where no wavelength conversion is used in the switch. This is equivalent to setting $w = 1$ in definition 1 and finding *the* largest non-overlapping bursts, $\mathbf{I} = \{\mathbf{I_1}\}$. Consequently, the following lemma can be deduced:

Lemma 2: The weighted CRLaW problem with no wavelength conversion is NP-complete.

We now consider two special cases for the CRLaW problem. Let us first define a new parameter called the *overlapping degree*, d_O. It is clear that in the above graph model, where burst overlapping is represented by edges between nodes, d_O will be equivalent to the nodal degree.

Lemma 3: If the overlapping degree of all n bursts within a contention region is the same, then the CRLaW problem can be efficiently solved using a polynomial-time algorithm with complexity of $\Theta(n)$.

Proof: Referring to Fig. 4.5(a), it is evident that when $d_O(B_i) = d_O(B_j)$ for every B_i and B_j in LaW, the CRLaW problem is reduced to simply finding the first d_{TS} shortest

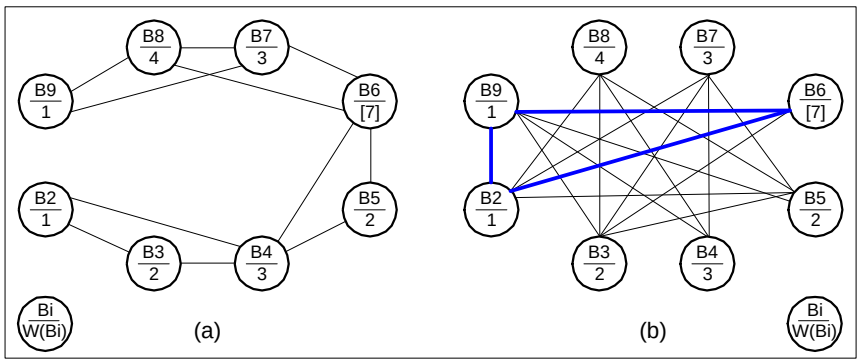


Figure 4.4. (a) Graph $\mathbf{G}$ representing the burst overlaps in example 1 (Fig. 4.2(b)); (b) graph $\overline{\mathbf{G}}$ and a Clique instance $\overline{I}_1 = \{B_2, B_9, B_6\}$. Note that graphs $\overline{\mathbf{G}}$ and $\mathbf{G}$ are inverted to each other.

bursts in the contention region. This is equivalent to implementing the shortest drop policy (SDP), which has the complexity of $\Theta(n)$. $\diamondsuit$

Another special case of the CRLaW problem deals with the case where the contention degree within the contention region for all time instances, TS, in the window is limited to one, i.e., $d_{TS} = 1$.

Lemma 4: For any given w, if the maximum number of bursts contending for w available channels is $w + 1$, an efficient algorithm with complexity of $\Theta(n)$ can be obtained.

Proof: In this case, in each contending time slot at *most* a single burst must be removed, as shown in Fig. 4.5(b). Hence, using the LCR shortest path algorithm, we can find the independent set with the *smallest* total weight, I_{opt}. The optimum contention resolution can be achieved by removing (dropping) all the bursts belonging to subset $I_{opt} = \{B_1, B2\}$. Note that the example in Fig. 4.2(b), also demonstrates such a case. $\diamondsuit$

4.3.2 LCR Performance

We now look at the performance boundaries of the LCR algorithm. Recall that in each iteration of the LCR algorithm, we attempt to find the smallest set of bursts which are involved in contention. The LCR iterations continue until all contention intervals (slots) are resolved; that is $d_{TS} = 0$ in each time slot TS. Clearly, the *optimum* solution to the LCR algorithm is obtained by removing *only* the excessive overlapping time intervals

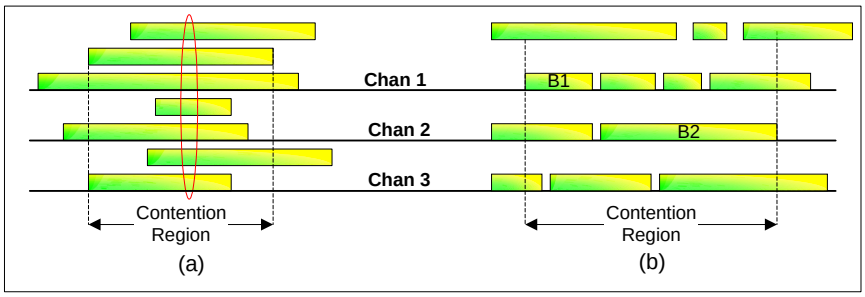


Figure 4.5. Spacial cases of the CRLaW problem: (a) overlapping degree for all contending bursts is the same (d_O =7); (b) number of available wavelengths is w =3 and the contention degree is $d_{TS} = w + 1$ =4.

(slots):

$$I_{opt} = \sum_{t=TS_o}^{TS_o+W} d_t, \tag{4.5}$$

where d_t is the contention degree and TS_o is the starting time slot of the burst window of size W.

Let us elaborate on the *upper bound* solution obtained by the LCR algorithm. We define the upper bound solution as the maximum number of bursts removed by LCR in order to resolve all contentions in LaW. Contention occurs when the number of overlapping bursts exceeds the number of available channels, w, by d_{TS}. Assuming that each shortest path iteration results in a set of non-overlapping bursts and that contention degree is reduced by exactly one unit after each iteration (the worst case consideration), the LCR algorithm will be equivalent to selecting a minimum-size set of *independent* bursts. In this problem, similar to CRLaW, bursts and overlap instances are represented by nodes and edges of graph $\breve{\mathbf{G}} = (\breve{\mathbf{V}}, \breve{\mathbf{E}})$, respectively. We denote this independent set of bursts by I_k for each iteration k having a size of $|I_k|$. In such a model, when contention occurs, the nodal degree, γ_i, of the node (burst) B_i will be equivalent to its overlapping degree. It follows that

$$\sum_{i \in I_k = \{B_i, B_j, \ldots\}} (\gamma_i + 1) = |\breve{V}_{k-1}|, \tag{4.6}$$

with $|\breve{V}_{k-1}|$ representing the number of existing nodes in the graph prior to each iteration k. For example, in Fig. 4.4(a) the total number of nodes (number of bursts in the window) is $|\breve{V}|$=9, $I = \{B_4, B_7\}$, $|I|$=2, γ_4=4 and γ_7=3.

Furthermore, in graph $\breve{\mathbf{G}}$, each *contending* node B_i will have a nodal degree γ_i of *at least* w, indicating that B_i is connected to at least w nodes, each of which, in turn, has a nodal degree of at least w. Consequently, Eqn. (4.6) can be expressed as

$$\sum_{n=1}^{|I_k|}(w+1) \leq |\breve{V}_{k-1}|. \tag{4.7}$$

By removing every contending burst (node) $B_i \in I_k$ with a nodal degree of at least w, as many as w nodes and $w(w+1)/2$ *overlap instances* between any two bursts (node pair) will be eliminated from graph $\breve{\mathbf{G}}$. For example, in Fig. 4.4(a), with $w = 2$, if B_3 is selected to be on the shortest path, $B_3 \in I_k$, then B_2 or B_4 could not belong to I_k, eliminating at least $w(w+1)/2 = 3$ overlapping instances or *edges*. We can express this as follow:

$$\sum_{n=1}^{|I_k|}\frac{1}{2}w(w+1) \leq |\breve{E}_{k-1}|. \tag{4.8}$$

Note that $|\breve{E}_{k-1}|$ represents the number of existing edges in the graph prior to each iteration k. Using Eqn. (4.7) - (4.8) it can be concluded that for each iteration k, maximum number of contending bursts eliminated, $|I_k|$, will be bounded by

$$|I_k| \leq min(\lceil\frac{|\breve{V}_{k-1}|}{w+1}\rceil, \lceil\frac{2\cdot|\breve{E}_{k-1}|}{w(w+1)}\rceil). \tag{4.9}$$

In the above expression, the maximum number of iterations will be equivalent to $1 \leq k \leq (d_O^{max} - \mathrm{w})$, where d_O^{max} is the maximum overlapping degree within the burst window. Also, we assume $|\breve{\mathbf{G}}| = |\breve{\mathbf{G}}_{\mathbf{I_o}}|$ and $|\breve{\mathbf{E}}| = |\breve{\mathbf{E}}_{\mathbf{I_o}}|$. We emphasize that the above expression is independent of burst lengths and their priorities.

4.3.3 Window Size Selection in LCR

A critical issue in the LCR algorithm is determining the window size, W. Clearly, in general, as the window size becomes larger, the LCR heuristic can perform more effectively. However, the trade-off will be forcing the bursts to be delayed by W time units per physical hop along the source-destination path. If the maximum burst length, L_{max}, is known, a reasonable choice for W will be $2 \cdot L_{max}$ to provide *full viewing* with regard to other incoming bursts in the window.

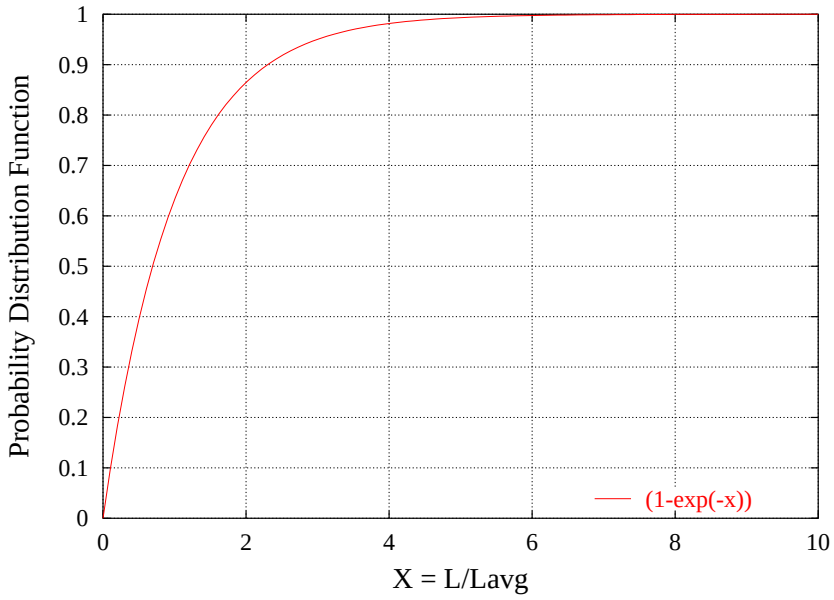


Figure 4.6. Probability distribution function of exponential distribution of burst length.

In a case that only burst length probability distribution function (PDF) is provided, W must be selected such that the probability of achieving full viewing is sufficiently large. For example, assuming the burst length is exponentially distributed with an average L_{avg}, the percentage of bursts whose length is longer than L is given by $1 - e^{-L/L_{avg}}$, which is plotted in Fig. 4.6. Therefore, if $W = 2 \cdot L_{avg}$, for a given contending burst, the probability of having full viewing will be about 85 percent.

4.4 Performance Comparison

In this section we present the simulation results for each of the introduced policies. We start by applying these algorithms to a simple example where packets can have low or high priority levels.

4.4.1 Numerical Comparison Between Different Contention Resolution Algorithms

Fig. 4.7 shows the expected incoming bursts from different ingress ports between time slots 19 though 33. We assume all bursts (B_1-B_8) have the same destination address and each

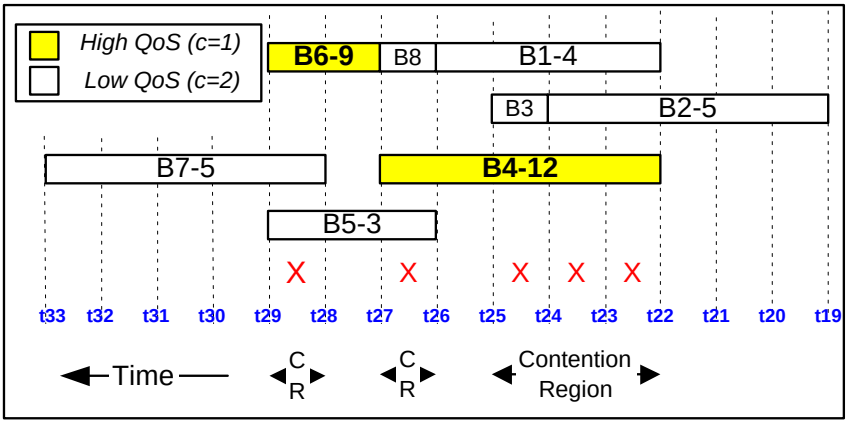


Figure 4.7. Example of incoming bursts (B_i); all bursts are directed to the same destination port, w=2, L_{max}=7, c_{max}= 2. Contention in slots, indicated by $\times$, occur when more than w bursts overlap.

outgoing port of the switch has only 2 wavelengths (w=2). First, we implement the LAUC-VF scheduling algorithm without utilizing any contention resolution algorithm. We assume high-priority data bursts, namely B_4 and B_6, were reserved much in advance. In this case the latest contending request will be dropped. Thus, discarding B_1, B_7, and B_8 can be considered as the worst-case outcome.

Using Segmentation, individual slots of B_2, B_3, B_5, and B_8 can be isolated and dropped. In this case, we ignore the impact of extra overhead requirements in order to divide data busts into independent segments.

We then consider the LCR algorithm with $W = 2 \cdot L_{max}$=14. In this case, the weights associated to the high-priority data burst B_4 and B_6 will be 7+5=12 and 7+2=9, respectively. The weights of low priority bursts are equivalent to their durations. Table 4.1 summarizes the resulting performance using different contention resolution algorithms. Note that LPD can potentially result in the worst-case performance while the SEG technique provides an upper bound on performance.

4.5 Simulation Results

In this section, simulation results are presented for each of the contention resolution schemes, namely the Latest Drop Policy (LDP), Shortest Drop Policy (SDP), Look-ahead Contention

Table 4.1. Performance results of implementing various contention resolutions for the example shown in Fig. 4.7.

Drop Policy	Bursts Dropped	Slots Dropped
LDP	B1, B7, B8	10
SDP	B1, B5, B8	8
LCR	B1, B5	7
SEG	B2(2), B3, B5(1), B8	5

Resolution (LCR), and Segmentation (SEG). These results are obtained with the following assumptions:

- The network consists of a single bufferless core switch with 8 input/output ports and each port consists of 16 data channels and a single control channel.
- Each link is bi-directional with a fiber in each direction and the transmission rate is 10 Gbps.
- Incoming IP packet lengths are fixed, 1250-byte, and the average number of IP packets in each data burst is 160: $L_{avg} = 160 \cdot 1250 = 200$KB.
- All data and control packet transmissions are slotted with the slot size granularity of 10,000 bytes. Hence, the average data burst duration is 20 slots.
- Wavelength conversion is utilized on all output ports of the switch.
- Data bursts can have three distinct priority levels, c =1, 2, 3, with distribution ratios of 10, 30, and 60 percent, respectively.
- Inter-arrival times between BHPs are exponentially distributed.
- Source-destination pairs are assigned based on a uniform distribution.
- Offsets between BHPs and their associated data bursts are fixed.
- The latest available unscheduled channel (LAUC) algorithm is adopted to schedule data bursts at the core nodes.

In our C-based simulation model, for each case of interest, the simulation was run until a confidence interval level of 90% was observed and an acceptably tight confidence interval (5%) was achieved. Calculations of the confidence interval were based on the variance within the collected observations [108].[1] All simulations were performed on a UNIX-based multiprocessor machine.

We represent the simulation results in terms of network load. We define burst blocking probability as the percentage of data bursts that were transmitted by the source edge node but never reached their destination node.

We first justify the slotted OBS transmission mechanism. Fig. 4.8 shows the blocking probability using the LDP algorithm with slotted and unslotted transmission. As the figure indicates, slotted LDP results in lower burst blocking probability that non-slotted case with variable length, particularly at lower loads. Clearly, the trade-off to having lower blocking probability using slotted transmission is its complexity and added delay.

Fig. 4.9 shows the overall performance of LCR compared to SEG and LDP algorithms. Note that, as expected, SEG and LDP provide the upper and lower bounds on performance, respectively. On average the LCR performs about 20 percent better than LDP. Note that, unless otherwise stated, $W = 2 \cdot L_{avg}$.

Fig. 4.10 suggests that LCR performs better than SDP in terms of burst blocking probability. This is due to LCR's deeper viewing ability. In a multi-switch system the overall blocking probability of the SDP algorithm can actually be increased due to its potential over-reservation. In such cases the performance difference between LCR and SDP, in terms of blocking probability, becomes more significant. Recall that a major issue with LCR is its per hop delay, which is equivalent to W time slots. The SDP can be considered as a reasonable tradeoff between reducing delay and slightly lowering the performance.

We now look at the performance of LDP and LCR when data bursts have multiple classes of service. Fig. 4.11 shows the resulting blocking probability for each of the three

[1] Refer to Appendix B at the end of the chapter for more information)

classes of service using LDP. We assume that length distribution and the average burst length, L_{avg}, are the same for all classes. In this case, the offset time of bursts with first and second class priority (c=1, c=2) will be $\Delta^{c1} = 6 \cdot L_{avg}$ and $\Delta^{c2} = 3 \cdot L_{avg}$. We assume that the offset for the lowest priority burst is much smaller than L_{avg} and hence, insignificant. Such offset assignment can provide a reasonable isolation of about 95.0% percent isolation between bursts with consecutive priorities. [109]. In our simulation we consider the offset for the lowest class bursts to be very small.

The results for blocking probabilities for each of the classes using LCR are shown in Fig. 4.12. Note that the blocking probability for each individual class of service improves when LCR is implemented in place of LDP. The performance results of the lowest two classes, $c = 2$ and $c = 3$, using LCR and LDP are compared in Fig. 4.13. Note that, regardless of burst priorities, the blocking probability can significantly be improved using LCR. As this figure suggests, load increase results in less significant difference between implementing LCR and LDP.

Similar performance results for multiple classes of bursts can be obtained when comparing SDP and LDP, as shown in Fig. 4.14. However, one major advantage of SDP is that there is no need to dedicate *extra* offset time to bursts with higher priority. Consequently, SDP can provide lower over-all end-to-end data burst delay when prioritized bursts are used. We emphasize that it is possible that, if the number of burst priority levels increases, for example $c_{max} > 3$, and the average hop distances is small, LCR can potentially provide lower end-to-end delay for high priority bursts.

The impact of the window size, W, in LCR is shown in Fig. 4.15. Note that as the window size increases, the burst blocking probability decreases. This figure suggests that the bulk of improvement is accomplished when $W \geq 2 \cdot L_{avg}$. This is consistent with our previous assumptions.

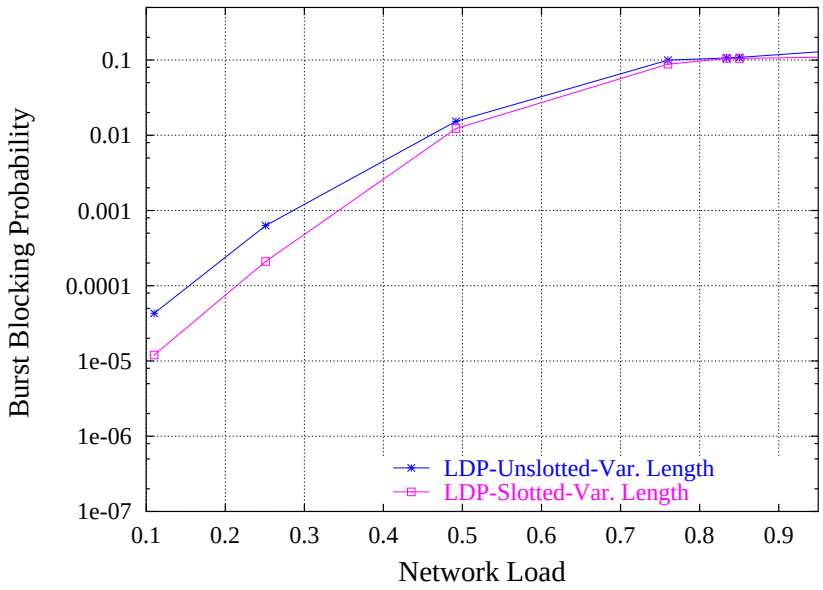


Figure 4.8. Comparing slotted and unslotted transmission using the LDP.

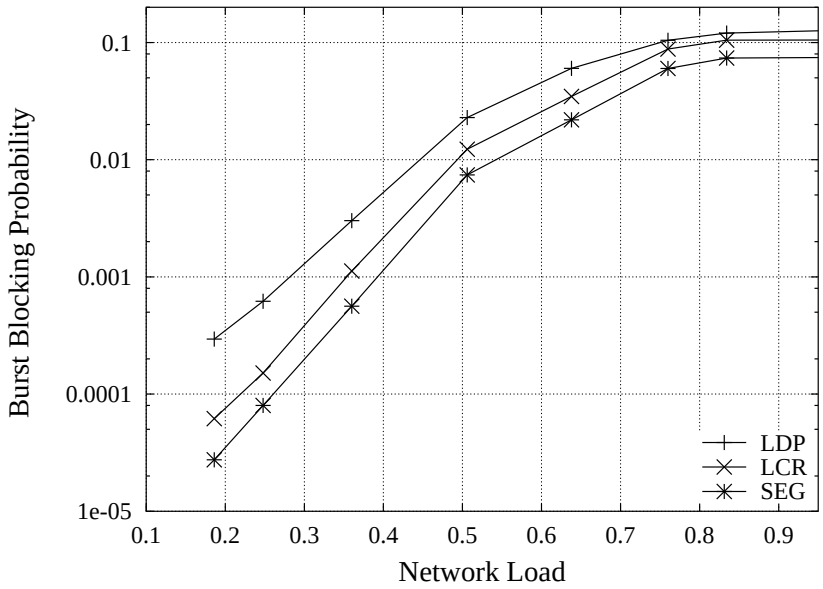


Figure 4.9. Overall BLR performance using different contention resolution schemes with W=40, L_{max}= 20 slots.

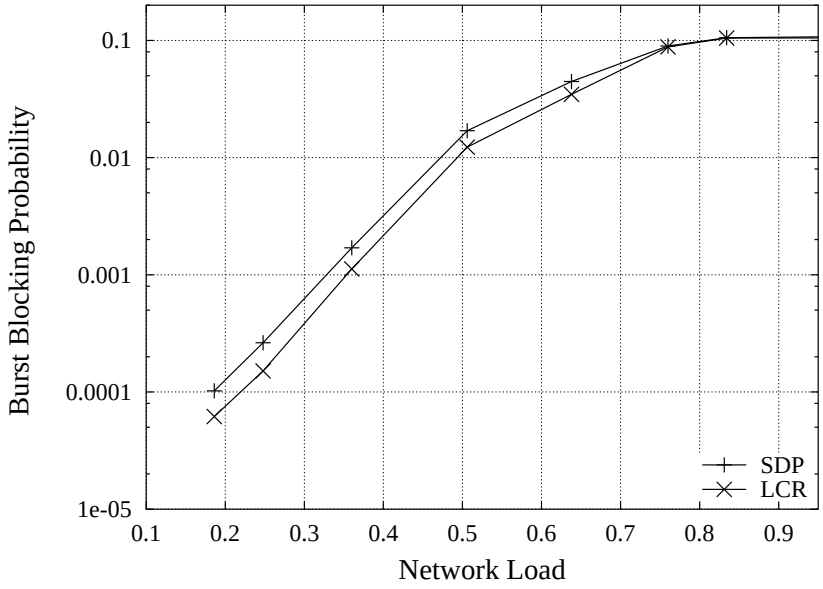


Figure 4.10. Overall BLR performance using LCR resolution schemes with W=40 slots and SDP.

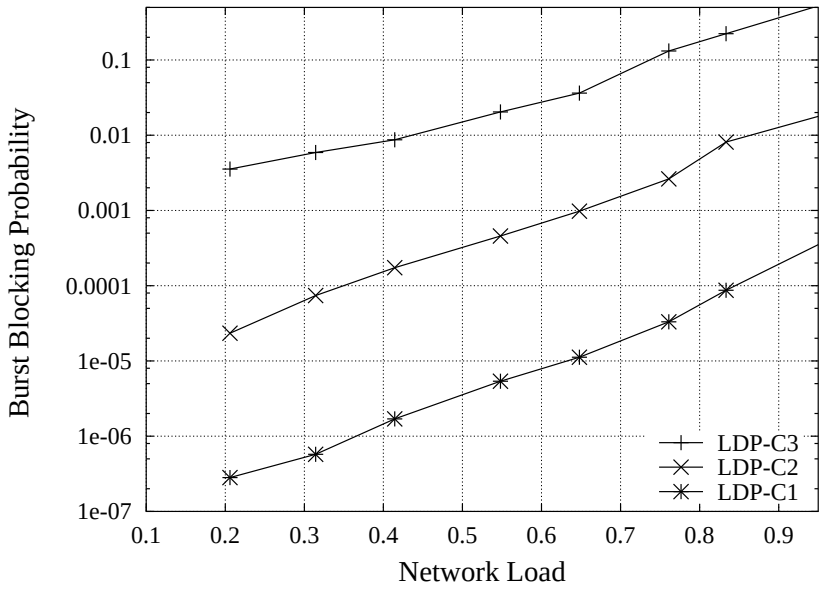


Figure 4.11. Burst blocking probability for all three classes using LDP; C1 indicates the highest priority level.

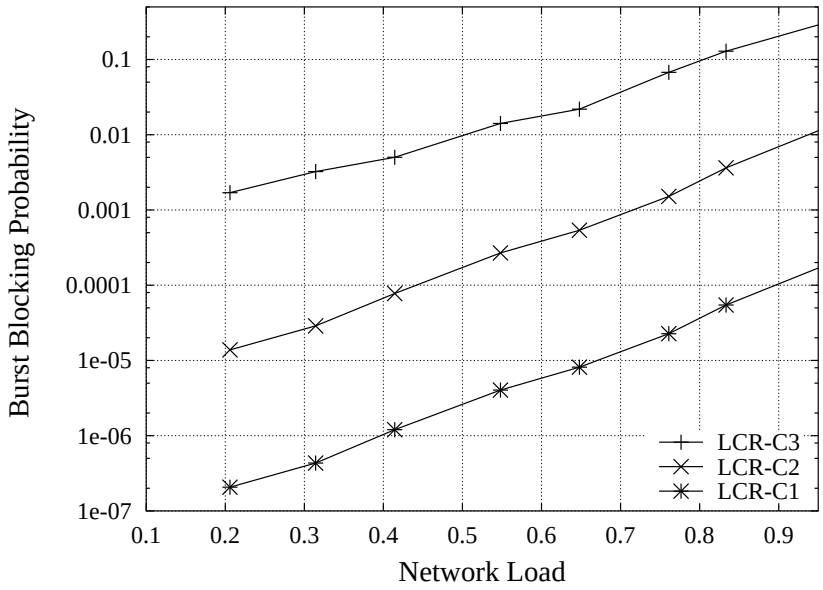


Figure 4.12. Burst blocking probability for all three classes using LCR; C1 indicates the highest priority level.

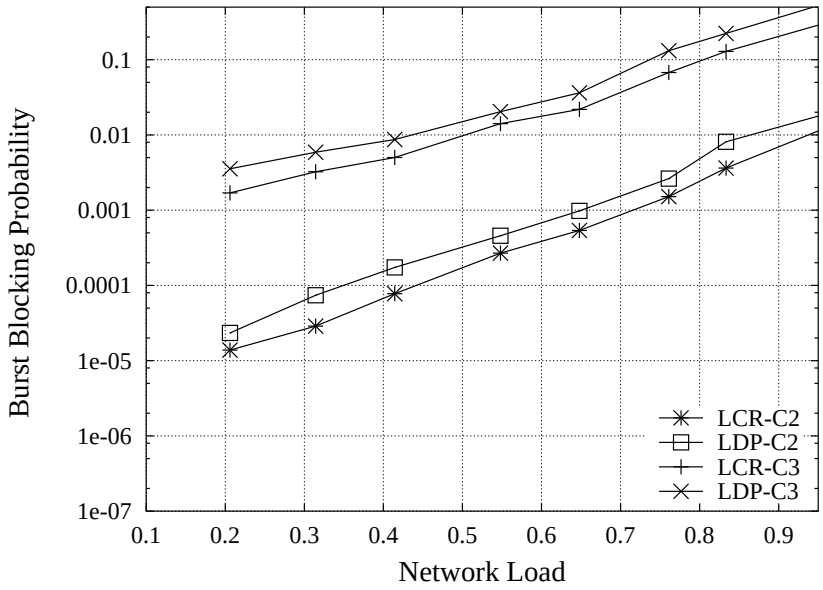


Figure 4.13. Burst blocking probability comparison of classes 2 and 3 in LDP and LCR.

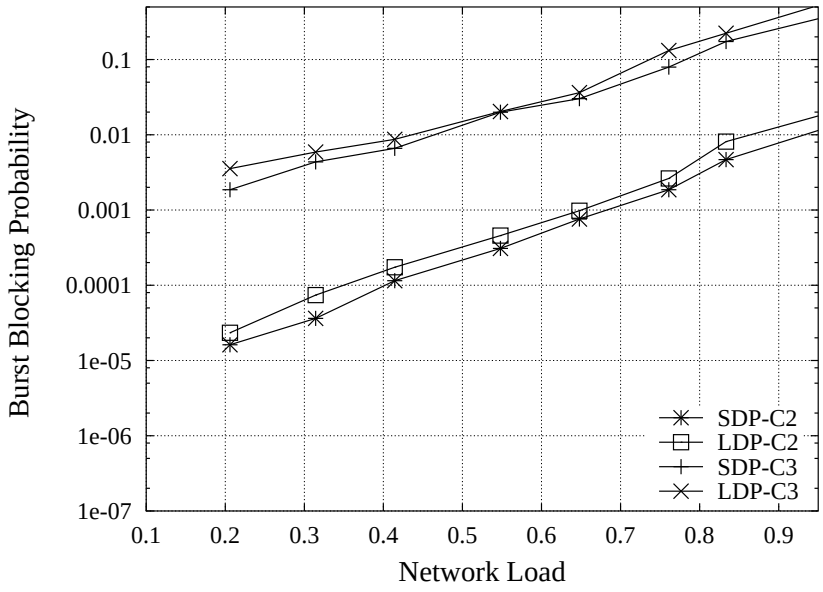


Figure 4.14. Burst blocking probability comparison of classes 2 and 3 in SDP and LCR.

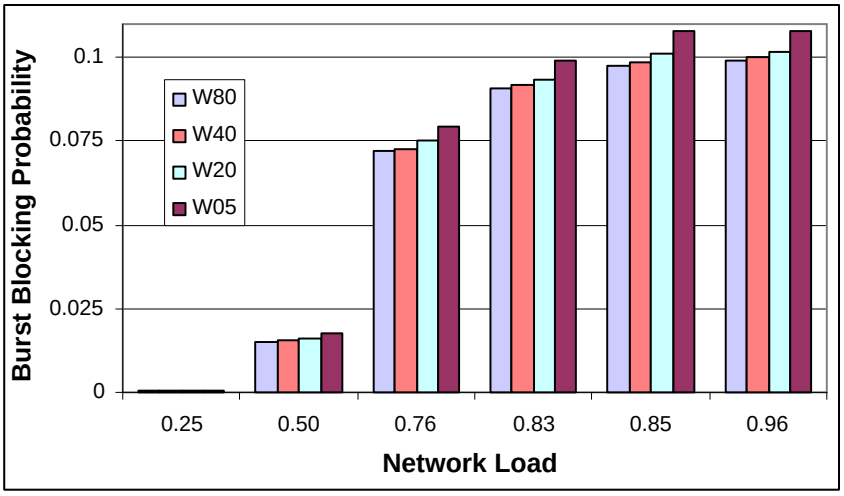


Figure 4.15. LCR overall performance with window sizes (W): 5, 20, 40, 80 slots with L_{avg}= 20 slots, indicated by $W05, W20, W40,$ and $W80$, respectively.

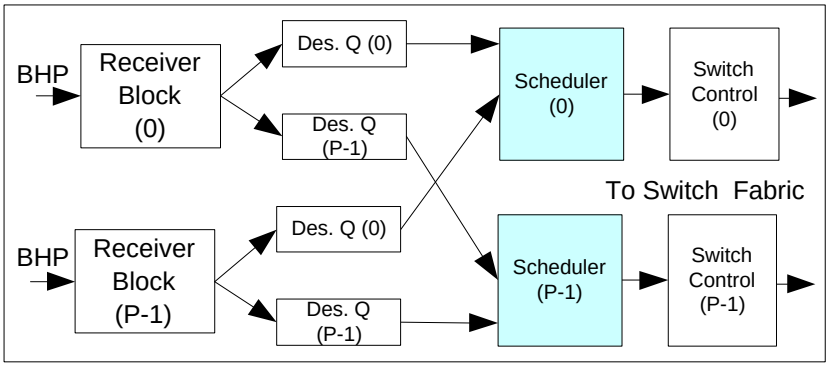


Figure 4.16. A distributed (parallel) architecture for the control packet processor (CPP).

4.6 Hardware Implementation

In this section we provide a general architecture for the control packet processor (CPP) unit in the core node, as shown in Fig. 4.1. Using this architecture, we implemented the shortest drop policy (SDP) contention resolution algorithm and evaluated its processing speed, scalability, and cost. We assume the core node has P ingress/egress ports each having w data channels and a single control channel. Without loss of generality, in our architecture we consider slotted transmission, in which both data and control channels arrive on slot boundaries. The data and control slots are assumed to have the same duration [68].

In our proposed *distributed* architecture, as shown in Fig. 4.16, each of P egress ports has a dedicated scheduler unit. After being wavelength de-multiplexed and converted from optical into electrical format, each incoming BHP, is processed through the BHP receiver block. The receiver block first checks each BHP for proper framing and parity information to ensure the packet validity. Then, upon detecting a valid BHP frame, the required information fields (such as destination, burst length, offset, QoS, ingress channel, etc.) are extracted for further processing.

The receiver block also generates a timestamp count, CNT, representing the control slot in which a BHP arrived. Therefore, each BHP is time stamped based on its corresponding data burst arrival time and the end of the data burst service time. The beginning of the burst arrival time (TS) is equivalent to the latest timestamp count value added to the offset time

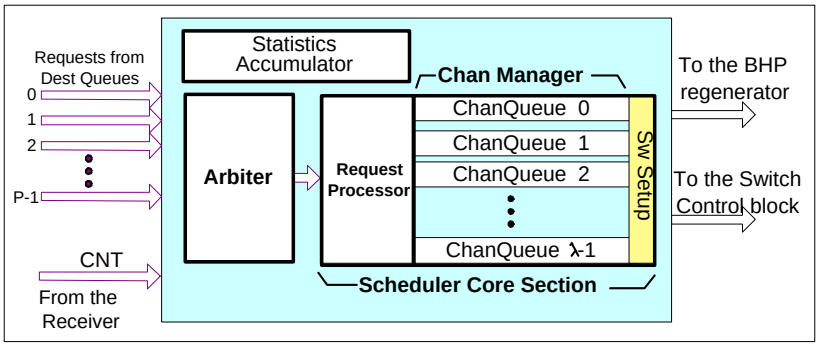


Figure 4.17. Details of the scheduler block and its interfaces, assuming P ingress/egress ports each having w data channels and a single control channel.

(Δ). The end of the scheduling time for data burst B_i, TE_i, is represented by

$$TE_i = TS_i + L(B_i) = (CNT + \Delta_i) + L(B_i), \tag{4.10}$$

Therefore, each request, BHP_i, requires a reservation from TS_i to TE_i and $L(B_i) = TE_i - TS_i$. The receiver block reformats each incoming BHP to include only the required payload and the scheduling timing information.

Depending on the BHP's destination, the receiver sends the reformatted BHP request to one of its P destination queues shown in Fig. 4.16. The destination queue is a prioritized digital queue that services requests according to the start of their scheduling time and their priority.

All destination queues with the same index (0 through P-1) are interfaced to the same scheduler block, each of which schedules requests for one of the egress ports on the switch fabric. Thus, requests with different destinations can be scheduled concurrently. Details of a generic scheduler block and its interfaces are shown in Fig. 4.17. The basic function of the scheduler block is to properly reserve sufficient resources for incoming BHP requests and their corresponding data bursts. Consequently, the contention resolution algorithm is the core part of the scheduler unit. In our architecture the scheduler block is divided into four hardware blocks: *arbiter*, *processor*, *channel manager*, and *statistics accumulator*, as shown in Fig. 4.17.

The scheduler's arbiter is required to control request flows into the scheduler. The arbiter ensures that requests with earlier scheduling time and higher priority levels are serviced first. When dealing with service differentiation, two critical issues must be addressed: possible service starvation for low-priority requests and fairness between requests coming from different control channels with the same destination [111]. Various arbitration schemes can be considered to ensure service differentiation. In our design, we implement a priority Binary Tree Arbiter (BTA) scheme to control the packet flow from destination queues into each scheduler. The BTA [112] is a Mealy machine in which decisions are made based on the previous states. Studies have shown that such arbitration schemes can provide fair allocation of resources among all requests while avoiding service starvation. Other advantages of BTAs include their fast processing time and scalability.

The arbiter passes the requests to the processor one at a time. Based on the request's scheduling time and its duration, the processor searches through all previous reservations on different channels and checks for any available bandwidth to accommodate the new request. If a request was successful, it is passed to the channel manager block. Obviously, an increase in the number of existing reservations for each channel results in longer search time per new request. If there are no available resources, the reservation request is denied and its incoming associated data burst is discarded. Note that different scheduling and contention resolution policies, such as LDP or SDP, can be implemented in the processor block.

The channel manager block contains as many as w channel queues (one for each channel on the egress port) and an update switch setup block. The processor sends an accepted request to the proper channel queue and the request is stored until its corresponding data burst is completely serviced. The storage time for a given request j in a channel queue is equivalent to $(\Delta_j + L(B_j))$ data burst slots. For simplicity, and without loss of generality, in our design, we assume that all incoming requests have constant offset. In this case, all requests will be sequentially stored in channel queues in order of their burst arrivals. Consequently, when the CNT changes, a new search for reservations with starting or ending

timestamps similar to the current timestamp count begins through all w channel queues.

Soon after a reservation request is reserved, the switch setup block sends a copy of the reserved request to the BHP Regenerator block, as shown in Fig. 4.1, for retransmission to the next core node. The scheduler block also includes a statistics accumulator block. This block can be used to keep track of the percentage of requests dropped and number of errors detected. Using a standard serial interface to the accumulator, each individual scheduler unit can be accessed and monitored.

4.6.1 Scheduler Prototyping

We develop a prototype module to implement the control packet processor block, as shown in Fig. 4.16. However, in the following paragraphs we mainly focus on implementation and performance of the scheduler unit. Each scheduler block, as shown in Fig. 4.17, is implemented using a reconfigurable hardware device, i.e., field programmable gate array (FPGA), which allows easy upgradeability. All sub-blocks in the scheduler unit are modeled using VHDL hardware descriptive language. The request processor sub-block was designed to support the shortest drop policy capable of handling multiple number of classes of service. The entire design functionality was tested and verified using the Cadence (NcSim) framework. The design synthesis was performed by Synplify. FPGA placement and routing was done using Quartus II provided by Altera [114]. The hardware model for the control packet processor protocol was targeted and optimized for an Altera APEX 20K FPGA family, namely EP20K400E, offering 2.5 million system gates and operating at clock rates up to 840 MHz. We assumed the operating frequency of the scheduler to be 500MHz.

4.6.2 Design Cost Analysis

We now analyze the memory cost of the scheduler units. We assume packet requests continuously arrive with no interruptions and the maximum offset time and maximum burst length are given as Δ_{max} and L_{max}, respectively. Each scheduler unit consists of w channel queues. The storage time for each request in the channel queue is equivalent to the

sum of its offset time, its duration, and the maximum processing time of the request in the scheduler. Since we assume the processing time is much smaller than a single slot time unit. In our analysis we ignore the processing time. Therefore, a request is maintained for ($\Delta_{max} + L_{max}$) time units (slots) until it is serviced. During this time period as many as $w \cdot P/L_{min}$ new requests can potentially arrive. Note that the worst case occurs when the length of each incoming burst is limited to a single slot, $L_{min} = 1$. Consequently, the total memory requirement for all w channel queues in the combined P schedulers, in units of bytes, will be

$$\begin{aligned} & w \cdot P \cdot [\frac{P \cdot w}{L_{min}}(\Delta_{max} + L_{max})] \cdot W_d \\ & = w^2 \cdot P^2 \cdot [\frac{\Delta_{max} + L_{max}}{L_{min}}] \cdot W_d. \end{aligned} \tag{4.11}$$

In the above expression, W_d is the width of each request (in bytes) stored in the channel queue. Eqn. (4.11) suggests, the cost of the scheduler increases exponentially as the number of channels per ports or number of ports on the switch increases.

4.6.3 Design Performance

We note that for practical reasons, the design we implemented in the FPGA device was limited for a single control channel per link and four ingress and egress ports, each having 16 channels. The results obtained for larger designs were simulated and verified using hardware simulation tools. For the purpose of this section, the main focus of our prototype was to examine schedulers' processing speed, scalability, and cost, using the shortest drop policy. In this experiment, all BHP requests were emulated using a series of random number generators embedded in the hardware.

Fig. 4.18 shows the number of cycles required to process each incoming request, when shortest drop policy is used as the contention resolution algorithm in the scheduler unit. Note that as the number of reservations stored in the channels queues becomes larger, the scheduling time for new requests increases. This is intuitive, since there will be more reservations to be verified. Once all channel queues have been saturated, the processing

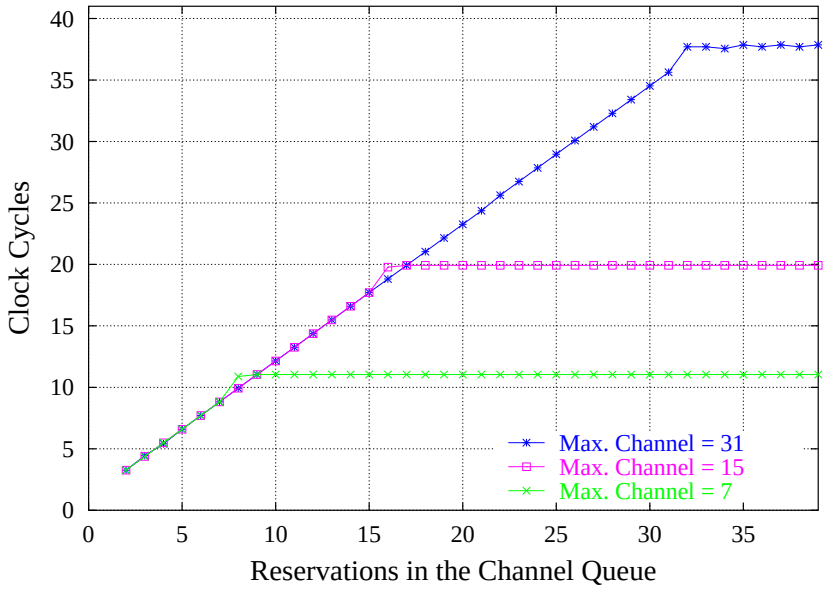


Figure 4.18. Number of clock cycles required for each new request to be scheduled on an egress port using the shortest drop policy.

time reaches a steady state. Fig. 4.18 also indicates that as the number of channels per port grows, the maximum number of cycles required for a new request to be scheduled increases.

Fig. 4.19 shows the actual hardware cost of implementing the scheduler block, as shown in Fig. 4.17, using the shortest drop policy after optimizing the design for EP20K400E Altera FPGA device. These results were obtained by ensuring that no request overflow occurs. Note that as more channels or ports are added, the cost of the scheduler, in terms of gate usage, exponentially increases. This observation is consistent with the results suggested by Eqn. (4.11).

4.7 Conclusion

In this chapter we presented several contention resolution algorithms for optical burst switching networks, namely the Latest Drop Policy (LDP), Shortest Drop Policy (SDP), Look-ahead Contention Resolution (LCR), and Segmentation (SEG). We discussed each algorithm and its implementation complexity and examined its performance in terms of burst loss probability for different classes of service. Simulation results show that the look-

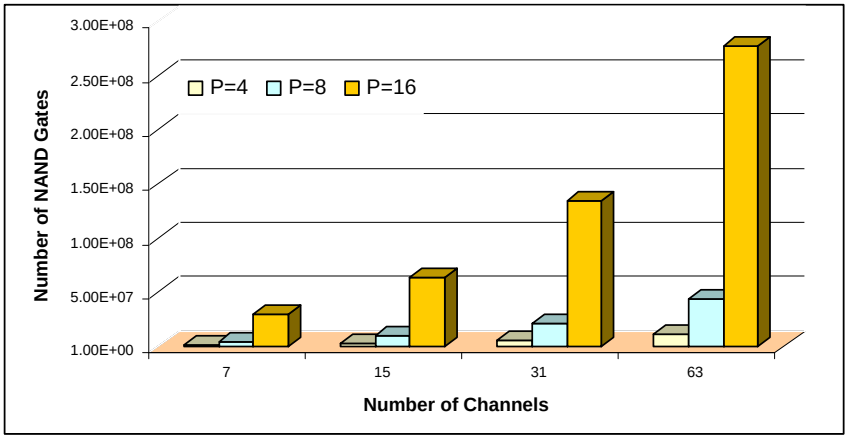


Figure 4.19. Hardware cost of the scheduler unit in terms of NAND gates for various number of egress ports and embedded channels.

ahead contention resolution algorithm can readily support service differentiation and offers high overall performance with moderate complexity. The LCR algorithm can be modified to reduce the total end-to-end burst delay at the cost of slightly lowering the performance.

We also presented a generic hardware architecture for fast BHP processing and discussed the design details of the BHP scheduler unit. Using this architecture, we implemented the shortest drop policy into a hardware programmable device such as FPGA. All blocks were modeled in VHDL hardware description language and the functionality of the design was verified using hardware simulations tools. We evaluated the performance of the scheduler in terms of cost, processing speed, and scalability when the shortest drop policy is implemented.

The results obtained in this study indicate that the shortest drop policy, as a special case of LCR, is an efficient tradeoff between reducing end-to-end burst delay and overall burst loss probability. Furthermore, the shortest drop policy is cost efficient in terms of memory requirements, highly scalable as the system size grows, and suitable for high speed operations.

One area of future work would be to extend the proposed look-ahead contention resolution to include limited buffering. Examining the fairness of the algorithm is also another important issue. For example, it is interesting to investigate whether LCR tends to favor

longer bursts and drop shorter bursts or, on average, treat all bursts similarly. Furthermore, we intend to use our proposed general hardware architecture for the scheduler unit such that it can replace the conventional event driven computer simulation. Under hardware simulation testbed a much deeper insight into the performance of the proposed scheduling and contention resolution algorithms can be achieved.

4.8 Appendix A: FPGA Implementation of the Scheduler

In this section we briefly describe the scheduler architecture and its detail design implemented on an FPGA device.

Fig. 4.20 abstracts the interface between the switch fabric, switch control unit, and the BHP scheduler in an OBS switch node. In this figure we assume that P physical links are connected to the switch node. Each incoming multi-channel links is demultiplexed into N data channels and Q control channels. All data channels are directly interfaced with the switch fabric and cut-through its pre-established paths as they arrive. Hence, the switch fabric has P ingress/egress ports, each of which contains N data channels. All control channels (a total of $P \cdot Q$) are initially converted into digital signals and processed in the BHP scheduler unit. The reservation requests, which are accepted in the scheduler unit, are passed to the switch control unit for proper switch setup.

4.8.1 BHP Scheduler

Details of the scheduler unit architecture are shown in Fig. 4.21. Each scheduler unit consists of ($P \cdot Q$) receiver blocks where BHPs are converted into digital signals and decoded. Information fields in BHP frames are extracted and converted into parallel data lines. Each input BHP is reformatted into reservation request packets (RRP) in order to include a time stamp and then passed on to proper classifier queue (CQ) based on the BHP's destination address.

The CQ consists of simple FIFO (first-in-first-out) devices in which RRPs with the same destination port are temporarily stored in order of their arrivals. The RRP priority level can be ignored while classifying them in different queue blocks for two reasons. First, each RRP in the classifier queue is serviced within a single clock cycle, which is negligible compared to the total processing time to schedule a packet. Second, packets with higher priority levels preempt previously scheduled requests and thus whether they are services first or not will not be a concern as long as we grantee their service prior to their data burst arrival.

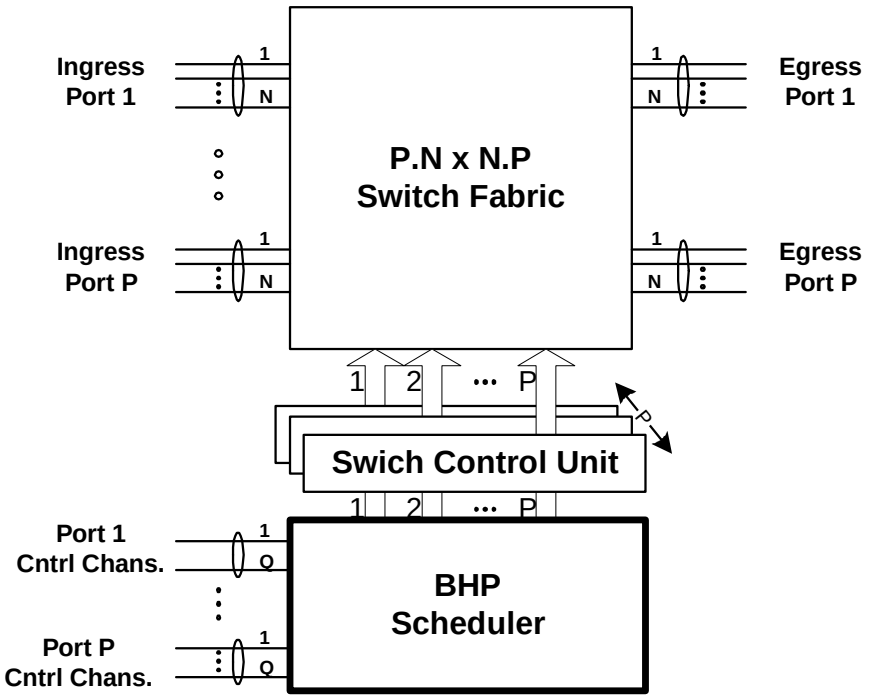


Figure 4.20. OBS switch node architecture.

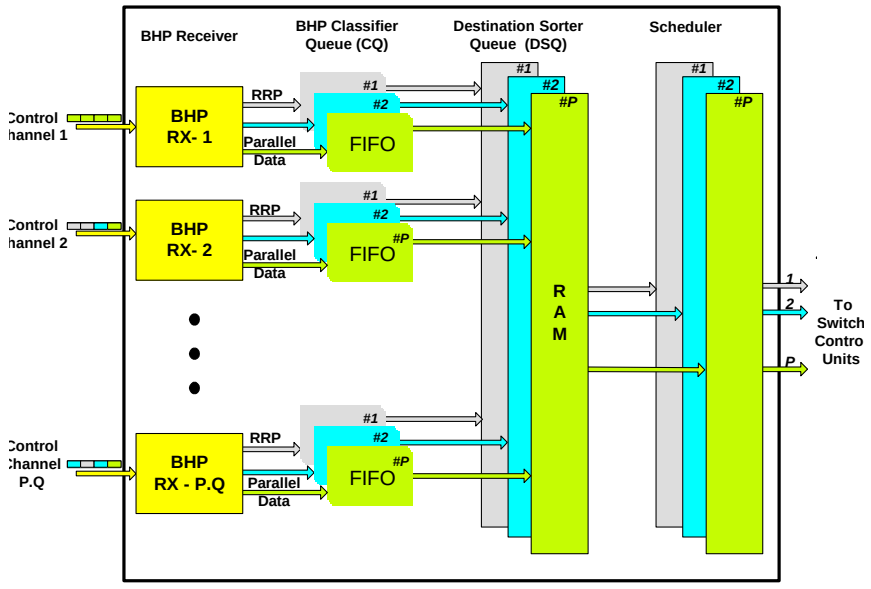


Figure 4.21. Parallel architecture for the BHP Scheduler in the OBS switch.

All CQs with the same destination id are interfaced with one of the P destination sorter blocks (DSBs). Similar to classifier queues, DSBs can also be simple storage blocks or more complex devices capable of performing sorting between RRPs with different priority levels. An arbitrator block (not shown in the figure) can be implemented to determine the order in which classifier queues can access the destination sorter. The arbitration scheme can be designed such that classifier queues with higher type of service requirement have access priority to the destination sorter. The arbitration algorithm must also address issues such as fairness and service starvation for lower priority RRPs. For simplicity, we consider a *round-robin* arbitrator, which grants access to one of the P CQs at a time. This architecture considerably simplifies the CQ's dimensioning criteria.

The actual scheduling algorithm resides in the scheduler block. Incoming RRPs are ordered based their type of service, data burst duration, and data burst arrival time. Each RRP request is examined and checked for available channel bandwidth. If there is no available resources the reservation request is denied and the RRP is discarded.

4.8.2 Switch Control Unit

The switch control unit contains a path set-up table, which indicates connections between ingress/egress channels. This information is updates at the beginning of each control slot. The egress channel is disabled when the reservation time is ended and the data burst passes through the switch fabric unless a new path is requested.

4.8.3 FPGA Implementation of the BHP Scheduler

The proposed parallel architecture, shown in Fig. 4.21, requires large amount of memory requirement. The actual size of each block directly relates to the size of the BHP frame and the number of information fields encoded in the frame.

In our implementation we focus on a bufferless switching system. However, the introduced scheduler architecture in Fig. 4.21 can support a burst switching system with data burst buffers. In this case only the scheduler algorithm needs to be modified to include

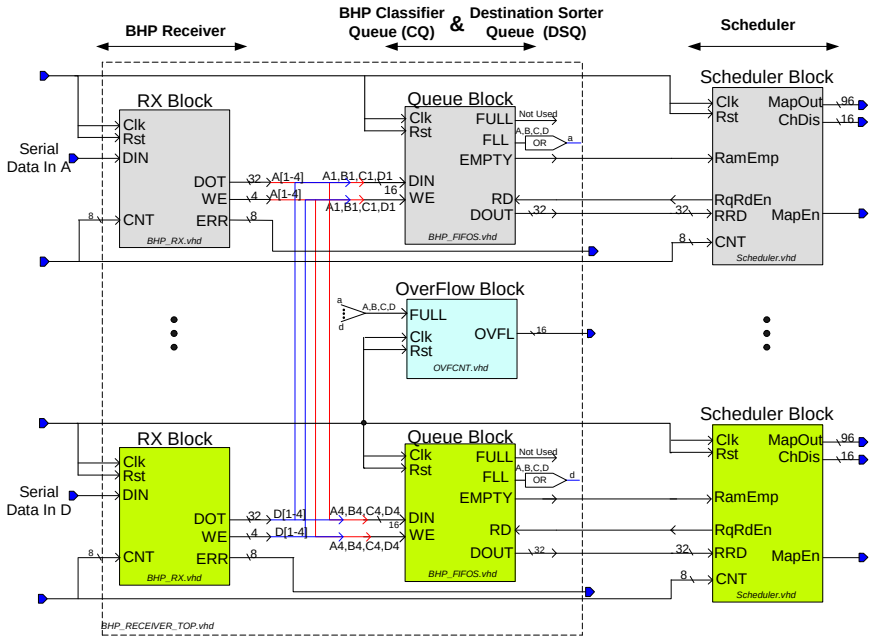


Figure 4.22. Detailed block diagram of the Scheduler FPGA.

variable arrival times for different data bursts depending on their buffering duration.

Furthermore, the presented architecture can handle service differentiation depending on the way packets with different priority levels enter into the Destination Queue and serviced. Thus, in the subsequent sections we only offer practical design alternatives to support service differentiation.

Fig. 4.22 shows a more detailed block diagram of the Scheduler FPGA representing the data flow through the scheduler block. This figure also identifies the actual high-level VHDL blocks and the way they are interfaced together. We assume that incoming BHPs are slot synchronized and all individual blocks operate on the rising edge of the clock.

In our design, each control slot is assumed to be 323.8 us. Containing as many as 32 BHP slots. The operating scheduler clock frequency was selected to be 50 MHz. Note that the clock rate was primarily based on processing ability of the scheduler. No tests have been performed to show the scheduler maximum tolerable clock frequency.

Each control channel, carrying BHPs, is interfaced with a scheduler block. Fig. 4.23 shows the Receiver block diagram of the scheduler. The main function of the receiver is to

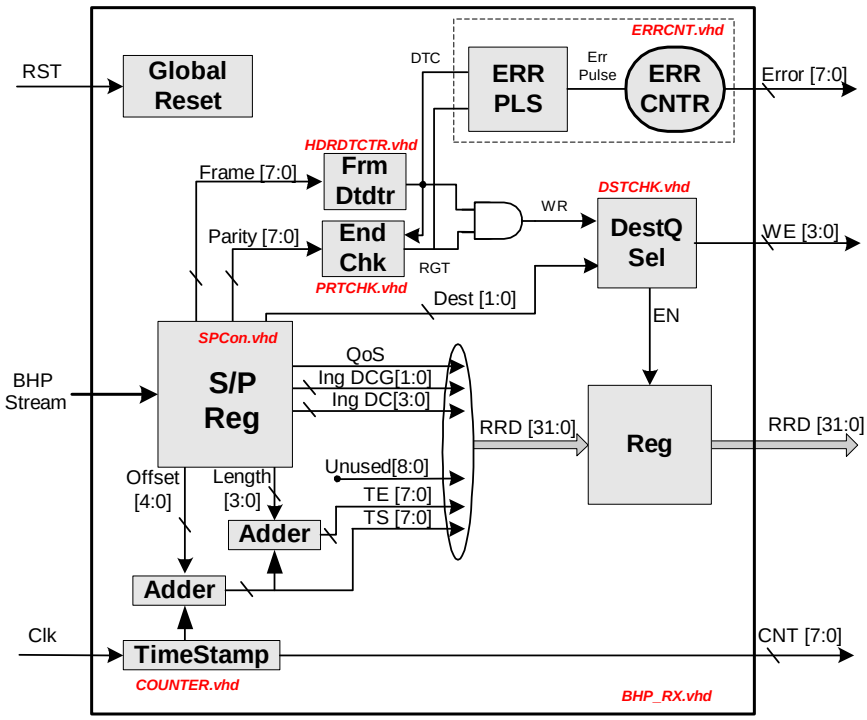


Figure 4.23. Details of the BHP Receiver block diagram shown Fig. 4.22 - xxx.vhd refers to the related VHDL code file.

convert the serial BHP streams into parallel data and to verify if they are error-free. Each BHP data stream is consisted of 34 bits, indicating its associated data burst characteristics such as its expected arrival time in terms of data burst slots, duration, destination port, etc. In the following paragraphs we describe the details of each building block in the scheduler FPGA.

Receiver Block

In the following sections we describe the functionalities of the Receiver sub-blocks.

Serial to parallel converter (S/P): In the design of the Receiver block we assume that the BHP body is transmitted in serial fashion. Hence, different fields embedded in the BHP must be extracted from the serial data, verified, and redirected to the proper sub-block.

Header Detector (Frm Dtdtr): The header detector block verifies the Header field of the incoming BHP serial data for a valid header pattern. If the valid pattern is detected the output DTC signal will be set to 1.

Ending Verifier (EndChk): A valid BHP body must have valid header and ending patters. Upon detection of the valid header, The Ending Verifier block checks the ending field of the incoming serial data for the ending pattern. The output RGT signal is set to 1 when a valid ending pattern is detected.

Error Counter (ERR CNTR): The Error Counter block counts the number pattern errors on each incoming BHP body. The counter increments if a header pattern is detected but no valid ending pattern is found. Errors are accumulated and can be reported to an external register.

Destination Identifier (Dest Sel): Assuming a valid header and ending pattern was detected on the incoming data, the destination value needs to be checked. The Destination Identifier reads the destination field on the BHP payload and notifies the proper queue block to accept the broadcasted data. The output of the Destination Identifier is a P-bit wide signal, which is connected to the write enable signals of queue blocks. Note that in this design we are assuming $P = 4$.

Time-stamp Generator (TimeStamp): The Counter in this block generates a count value ranging from 0 to 255. Each count value (CNT) represent as many as 16190 clock cycles. This number corresponds to the length of the control slot (323.8 us) and the system clock frequency. Each incoming BHP is time stamped with two arbitrary 8-bit values ranging from 0 to 255. The Start Time (TS) identifies the expected arriving slot of the DB corresponding to the current BHP and it is equivalent to

$$TS = Current\_CNT + Offset.$$

In this design we assume that the offset value is constant and expressed in terms of time slots. On the other hand, The End Time (TE) determines the relative slot in which the data burst's service has been completed. This time is equivalent to

$$TE = TS + Burst\_Length = Burst\_Length + Offset\ .$$

Where burst lengths are expressed in terms of time slots.

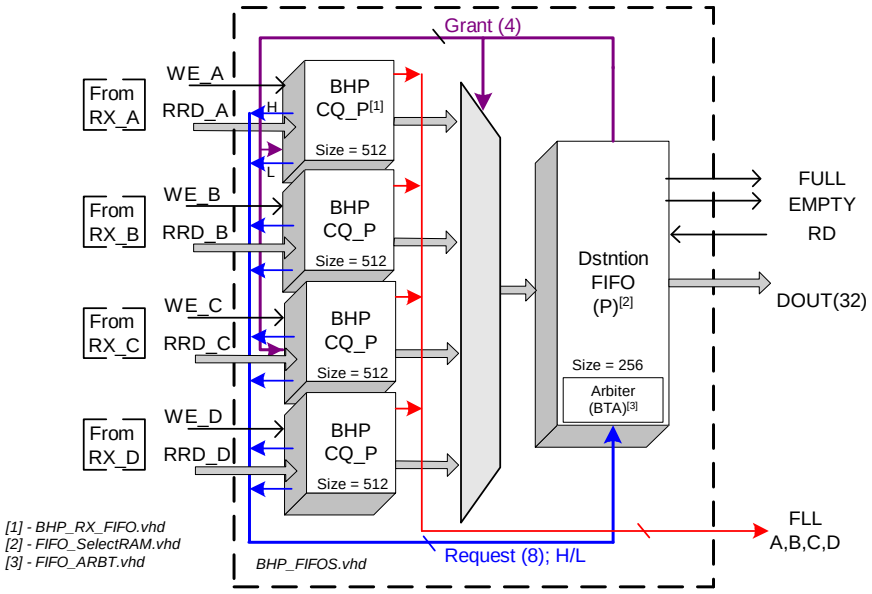


Figure 4.24. Queue block diagram.

Queue Block

Queue blocks combine the functionalities of BHP classier queues and destination sorter queues, as shown in Fig. 4.21. Each Queue block consists of $P = 4$ Classifier Queue (CQ), a single Destination Queue (DQ), and an arbiter. The total number of Queue blocks is equivalent to the number of egress ports, which in turn corresponds to the number of Receiver and Scheduler blocks. RRD packets from Receiver blocks with the same destination port are directed to the same Queue block. Fig. 4.24 shows the details of the Pth Queue block. Each of P CQ RAM blocks receives RRD packets from its dedicated receiver and sends a high or low request to the arbiter block. The request type depends on the QoS value of the RRD. In our design only two levels of priority is supported by the CQs. At each clock cycle the Binary Tree Arbiter (BTA) selects which CQ can transmit its data to the DQ based on its requested priority-level.

Scheduler Block

The heart of the BHP Scheduler unit is the scheduler block. As we explained in Section 4.6, the basic functionalities of the scheduler block are (1) deciding to reserve or discard

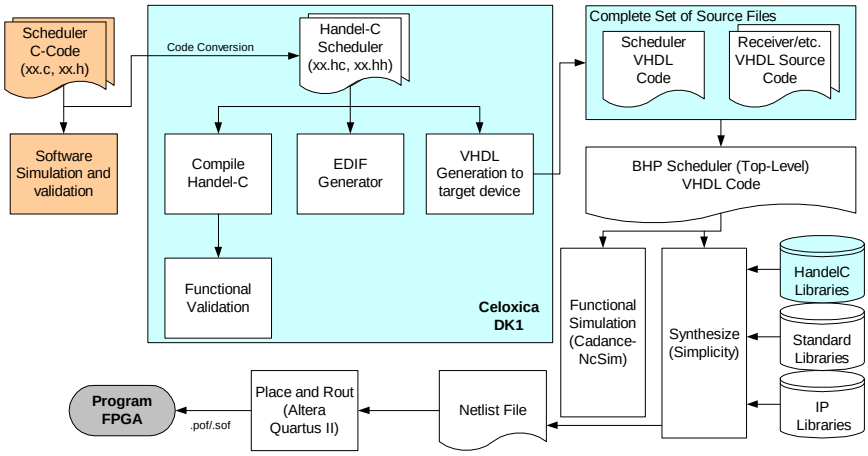


Figure 4.25. Hardware target design flow of the scheduler block.

the incoming request packets (RRDs) and (2) establishing a register map, which specifies how the switch fabric should be set up.

Tools and Methodology

An overview of the design methodology used to develop the BHP Scheduler FPGA is presented in Fig. 4.25. This synthesis-based design flow has four main steps: Design Creation/Verification, Design Implementation, and Programming. Interested reader can refer to [113] for more information.

4.9 Appendix B: Result Accuracy

There are three kinds of lies: lies, damned lies, and statistics;

Benjamin Disraeli British politician (1804 - 1881).

In general, most performance measurements result in a random quantification of the performance. Hence, if the measurements were to be repeated, the results would *not* be exactly the same. For instance, different sets of randomly generated data bursts can result in different burst blocking probabilities. Every performance measurement has a mean and variance. Means alone are not enough to identify the correctness of the performance measurements. For example, if we measure the probability of burst loss probability in an OBS network 100 times using only 1000 bursts, the performance results in all cases may differ significantly. This implies that the variance of our measurements with 1000 packets as our sample space, is very large. On the other hand, experimenting with infinite number of data bursts (the entire population) is clearly impossible. Hence, we like to achieve an *estimate* to population characteristics (such as population mean, μ).

One way to ensure the results are accurate and the sample space is large enough (sufficient number of data bursts are generated during the run time) is to achieve probabilistic bounds on the performance measurements. In this case, we may be able to get two bounds, C_1 and C_2, such that there is a high probability, $1 - \alpha$, that the mean is in the interval (C_1, C_2):

$$Probability\{C_1 \leq \mu \leq C_2\} = 1 - \alpha. \tag{4.12}$$

The interval (C_1, C_2) is called the *confidence interval* for the population mean, α is called the *significance level*, $100(1 - \alpha)$ is called *confidence level*, and $1 - \alpha$ is called the *confidence coefficient*.

Fig. 4.26 shows an example of a performance measurement, namely, burst blocking probability. In our simulations, the run time is set to achieve a confidence interval of 5% or less at 90% confidence level. The number of times we run the experiment to measure the mean is usually set to 50.

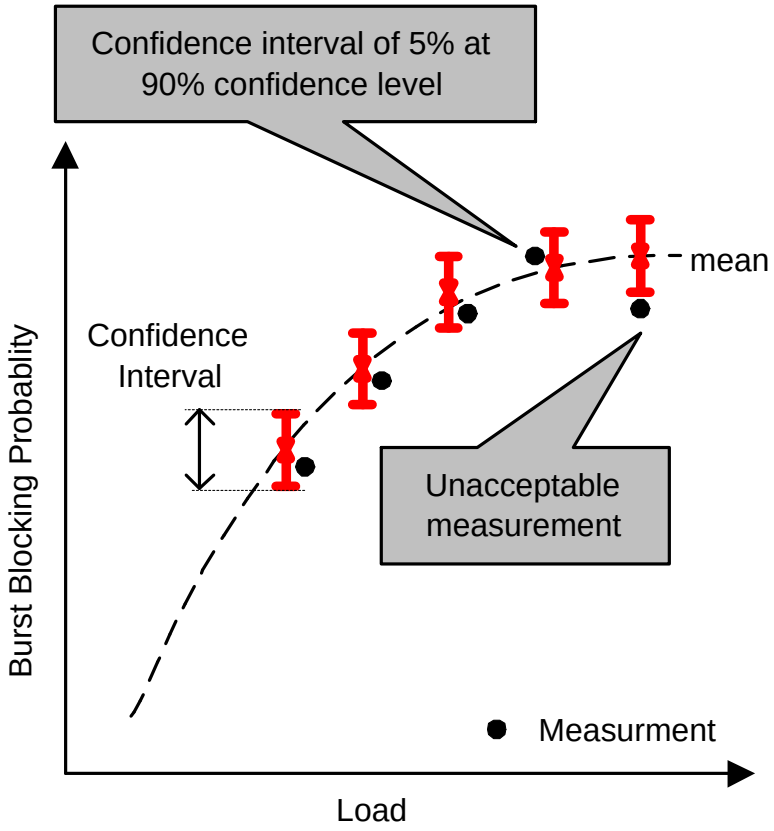


Figure 4.26. Example of confidence interval.

CHAPTER 5

A FEEDBACK-BASED CONTENTION AVOIDANCE MECHANISM FOR LABELED OPTICAL BURST SWITCHED NETWORKS

5.1 Introduction

A major concern in OBS networks is high contention and burst loss due to output data channel contention, which occurs when the total number of data bursts going to the same output port at a given time is larger than the available channels on that port. Contention is aggravated when the traffic becomes bursty and when the data burst duration varies and becomes longer. Contention and loss may be reduced by implementing contention resolution policies, such as time deflection (using buffering [101], [116]), space deflection (using deflection routing [103], [117], [102], [118], [118]), and wavelength conversion (using wavelength converters [119]). When there is no available unscheduled channel, and a contention cannot be resolved by any one of the above techniques, one or more bursts must be dropped. The policy for selecting which bursts to drop is referred to as the *dropping policy* and is used to reduce the overall burst dropping probability and consequently, to enhance link utilization. Several dropping policy algorithms have been proposed and studied in earlier literature, including the shortest-drop policy [68], segmentation [92], and look-ahead contention resolution [67].

As shown in Fig. 5.1, the above contention resolution policies are considered as *reactive* approaches in the sense that they are invoked after contention occurs. An alternative approach to reduce network contention is by *proactively* attempting to avoid network overload through traffic management policies. Consequently, contention avoidance policies attempt to prevent a network from entering the congestion state in which burst loss occurs. An ideal contention avoidance policy must serve several concurrent objectives: minimize the throughput, minimize the average end-to-end packet delay, operate with minimum additional signaling requirements, and guarantee fairness among all users.

In general, contention avoidance policies can be implemented in either *non-feedback-based* or *feedback-based* networks, as shown in Fig. 5.1. In a non-feedback-based network, the ingress nodes have no knowledge of the network state and they cannot respond to changes in the network load. Therefore, without requiring any additional signals in the control plane, each node regulates its own offered load into the network through traffic shaping (e.g., forcing the data bursts to enter the OBS network at a regulated rate) or traffic rerouting and load balancing based on a predefined traffic description. One way to perform the traffic shaping is through a burst assembly mechanism such as the ones proposed in [48], [49], [95], [120]. In [121], the authors propose regulating data bursts by combining periodic traffic reshaping at the edge node and a proactive reservation scheme. Traffic rerouting on alternative shortest paths (or load splitting) can also be implemented as a way to reduce link contention. The main challenge in implementing the contention avoidance policies in non-feedback-based OBS networks is to define the traffic parameter, such as peak rate and average rate at each edge node, in order to avoid or minimize link contention.

In a feedback-based network, one way to avoid contention is by dynamically varying the data burst flows at the source to match the latest status of the network and its available resources. Thus, as the available network resources are changed, a source should vary its offered load to the network, accordingly. Two critical issues in any network with feedback mechanism are determining what type of information must be conveyed to the source *and* interpreting the conveyed information and reacting to the current network state. We refer to these issues as *signaling strategies* and *control strategies*, respectively.

In the past two decades, numerous studies have been dedicated to designing and analyzing contention avoidance (or congestion control) mechanisms in TCP and ATM networks. Many different protocols have focused on the signaling strategies. For example, in congestion control approaches, such as DECbit [122], and its variations including Selective Binary Feedback (SBF), Random Early Detection (RED) [123], Explicit Congestion Notification (ECN) [124], and Proportional Control Algorithms (PRCA) [125] a single bit in the packet header explicitly notifies the source about the congestion in downstream nodes.

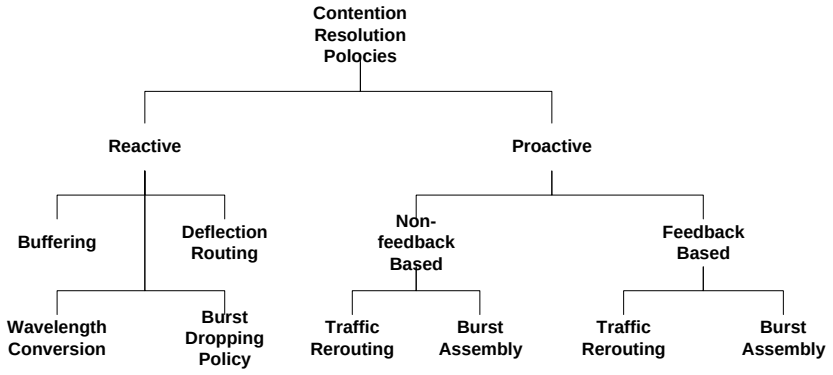


Figure 5.1. Categorizing different contention resolution mechanisms.

Other feedback protocols such as eXplicit Control Protocol (XCP) [135], and Available Bit Rate (ABR) [126] utilize multi-bit feedback messaging, which explicitly indicates the degree of congestion at the bottleneck.

There are also many proposals which focus on adjustment algorithms including Binomial congestion control algorithms [127] and [128], Additive-Increase Multiplicative-Decrease (AIMD) [129], Multiplicative-Increase Multiplicative-Decrease (MIMD) [127] and [128].

A majority of existing feedback protocols use end-to-end congestion avoidance mechanisms. Therefore, explicit or implicit information about the status of network are passed to destinations nodes and then returned to the source. Several studies, including [130], have considered hop-by-hop control mechanisms in which feedback information is exchanged between the neighboring nodes. Thus, each node adjusts its own rate to match its adjacent nodes.

A number of feedback-based contention resolution schemes have been proposed for OBS networks. One way to avoid contention in feedback-based OBS networks is to reroute some of the traffic from heavily loaded paths to under-utilized paths [70]. In this case, a core node sends feedback messages containing the load information of its overloaded output links to the ingress nodes. A similar approach has also been introduced by [73] where the authors consider balancing the data burst traffic between predefined alternative paths.

Another way to avoid contention is to implement a TCP-like congestion avoidance mechanism to regulate the burst transmission rate [60], [91]. In this approach, the ingress edge nodes receive TCP ACK packets from egress edge nodes, calculate the most congested links, and reroute their traffic accordingly. A potential drawback of these schemes is that rerouting the data bursts to alternative paths can potentially cause link congestion elsewhere and thus result in possible network instability. Furthermore, when the round trip delay is large and the network operates at a very high speed, the edge nodes' responses to the network change tend to be slow. In [131], [132] the authors propose a hop-by-hop feedback mechanism. In this approach, called Backward Explicit Congestion Notification / Congestion Restoration Notification (BECN/CRN), when congestion occurs, intermediate node sends BECN signal to its neighboring nodes. The neighboring nodes, based on the received BECN signals, decide how to deflect outgoing bursts on an alternative route.

In this chapter we propose a new rate-based congestion avoidance mechanism for bufferless OBS networks where multi-bit explicit feedback signaling is sent to each edge source node indicating the required reduction in the burst flow rate going to the congested link. We refer to such feedback-based contention avoidance as proportional control algorithm with explicit reduction request (PCwER). In this scheme, during the underload periods, the rate of transmission increases additively (AI), whereas during congestion period, the sending rate decreases multiplicatively (MD). Our proposed contention avoidance mechanism utilizes OBS network characteristics, and it differs from previous proportional rate-based algorithms based on the following four assumptions: (1) feedback information reflects the actual load level (or loss rate) at the congested link; (2) there is no queuing delay on intermediate nodes, and link propagation delays are known to all nodes; (3) the feedback signal *specifically* notifies the source by how much it should reduce its rate to match the targeted congestion level of the network; (4) the feedback signal is transmitted to the source from the bottleneck switch node, rather than the destination nodes as in end-to-end contention avoidance mechanisms.

The architectural details of our proposed feedback mechanism in OBS networks are

also described in this chapter. We explain how feedback signals can be framed within label-switched OBS networks. Through a simple fluid model we analyze convergence and evaluate the fairness of PCwER. In addition, by means of simulation, we examine the performance of the PCwER contention avoidance mechanism under specific network conditions. We compare our results to the case without source traffic control in terms of blocking probability and network throughput. We show that our approach behaves well in practice and responds quickly to any change in network status, while improving the overall network performance.

The rest of this chapter is organized as follows. In Section 5.2, we elaborate on the main components of a general feedback-based contention avoidance mechanism. In Section 5.3, we briefly describe the basic blocks and architecture of the label-switched feedback-based OBS network. In Section 5.4 we elaborate on details of our proposed contention avoidance algorithm. In Section 5.5, we analyze behavior of PCwER. Finally, in Section 5.6 we present performance results obtained by means of simulations, followed by concluding remarks in Section 5.7.

5.2 Feedback-Based Congestion Control Components

Typically, in TCP/IP or ATM networks, a traffic source must control its transmission rate in response to the receiver state as well as the network state [111]. However, in OBS networks, it is generally assumed that the ingress and egress nodes have adequate buffering capacity, and that matching the source rate to the service rate at the destination is not of great importance. Henceforth, the main objective in feedback-based contention avoidance schemes in OBS networks is to dynamically adjust (or regulate) the data burst transmission rate at edge nodes in response to core nodes' feedback signals such that network overload is avoided or minimized. We refer to such closed loop traffic regulation as *admission control*. The schemes that determine the way the traffic is regulated are called *admission control strategies*. Fig. 5.2 identifies two key elements in feedback-based contention avoidance schemes in OBS networks: control and signaling strategies. The feedback control strategy

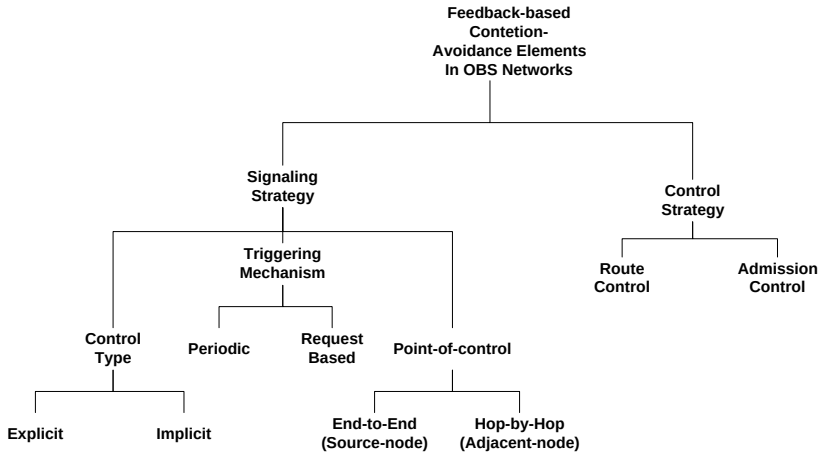


Figure 5.2. Control elements in a feedback-based contention avoidance scheme.

refers to the type of action that the node receiving the feedback messages performs. For example, an edge node can reduce the transmission rate through admission control strategies or reroute data burst flows going through the congested link. On the other hand, the feedback strategy indicates how the current state of the network is measured and is communicated to other nodes (such as ingress or egress edge nodes or intermediate core nodes). The feedback signaling strategy involves the following taxonomies:

(a) Feedback control type: refers to the type of the control messaging that is used to communicate the current state of the network to the source. The signaling type can be *explicit* or *implicit*. In the former, the feedback signal explicitly indicates the congestion state and the requested transmission rate (or transmission rate reduction). In the latter, the feedback signal indicates the rate of the packet loss on a particular link or in a node.

(b) Feedback triggering mechanism: indicates how often the feedback signaling is sent to upstream nodes. For example, the feedback signals can be transmitted periodically or based on some other node's request. Once the feedback signal is triggered it can be *broadcasted* to all sources or sent to particular nodes.

(c) Feedback point-of-control: refers to the nodes which respond to the feedback messages

and take action to avoid congestion occurrence. The responding nodes can be the edge nodes or the adjacent core nodes. We refer to these as end-to-end and hop-by-hop signaling, respectively.

In this section we only focus on a feedback-based contention avoidance mechanism in which each core node periodically broadcasts explicit link information to all edge nodes requesting them to dynamically adjust their data burst transmission rate if necessary. Thus, upon receiving the feedback information, edge nodes invoke their admission control and reduce the transmission rate of data burst flows passing through the congested link according to the requested rate. Note that all bursts belonging to the same burst flow share identical source and destination nodes. The admission control strategy we adopt in our study is a leaky bucket-based approach in which data bursts are scheduled on available wavelengths and transmitted according to a sustainable rate governed by feedback transmission rate reduction requests from intermediate nodes. In proportional control algorithm with explicit reduction request (PCwER) the total volume of offered traffic is not changed; rather only the transmitting rate of the data burst flow is regulated through the admission control. The regulated traffic rate (bursts/sec) is directly related to the state of the congested link. Once a link is over-utilized, the reduction in the transmission rate continues until the core node clears out the congestion condition. At this point, the edge node attempts to resume its original transmission rate according to some ramp-up policy such as incremental rate increase.

5.3 Network assumptions and Node Architecture

Detailed architectural design of core and edge nodes are provided in [100], [68]. In this section, we assume the OBS network under discussion consists of $|N|$ core nodes and $|L|$ links represented by sets $N = \{1, 2, .., n\}$ and $L = \{(1, 2), ...(j, k)\}$, respectively, where $j, k \in N$. Each link is characterized by the number of wavelength channels it carries, W, and the capacity of each channel, S. In addition, we assume each core node is connected to one or more edge nodes. Each edge node determines the source-destination route, $\mathbf{R}(\mathbf{s}, \mathbf{d})$,

and has sufficiently large buffers in order to store incoming packets due to network congestion and transmission latency. On the other hand, switch nodes are bufferless and hence, upon link congestion, data bursts will be dropped. Furthermore, we assume that each intermediate node n knows the set of source nodes that are contributing to the traffic load on an egress link (j, k), $\Lambda^n_{j,k}$, and all nodes have full knowledge of propagation delays between each source-destination node pair, $T(s, d)$.

Without loss of generality, we consider label-switched OBS networks using a Generalized Multi-Protocol Label Switching (GMPLS) control plane [133], [116]. In this model, the transmitted bursts are routed through individual Label Switch Paths (LSPs). We assume that the intermediate core nodes have no buffering capacity, and that incoming LSPs can either cut through the core nodes or be blocked. When the measured load on an egress port exceeds a predefined load threshold, the congested core node sends back a *flow-rate reduction request* (FRR) signal to ingress edge nodes requesting them to reduce the transmission rate of LSPs sharing the congested link. The feedback signaling to the source nodes can be implemented using the Label Distributed Protocol (LDP) employed in GMPLS. In this case, the feedback reduction request messages will be similar to the NACK message and will include the following information:

<LSP Label, Core Switch Address, FRR>.

The value of FRR indicates the actual rate reduction value required by the switch on link (j, k). The core switch address is provided in case the ingress edge node was allowed to use an alternative path for transmitting the affected LSP. The actual feedback messaging can also be implemented via Resource reSerVation Protocol (RSVP) [111]. In this case the FRR messages are encapsulated into the RESV messages propagating to upstream nodes. It must be noted that the feedback signaling can also be deployed independent of the RSVP or LSP control planes. In the rest of this section we refer to an LSP and a burst flow interchangeably.

5.4 OBS Rate-based contention avoidance algorithm

The basic idea in the proportional control algorithm with explicit reduction request (PCwER) is that each core node calculates the load on each of its egress links (reflecting loss probability) and reports that to edge nodes. Based on received feedback information, each source varies its transmission rate.

5.4.1 Signalling Strategy

In the PCwER contention avoidance mechanism, each core node maintains the load information on each of its egress link, (j, k), denoted by $\rho_{j,k}$. One way to calculate the load is to measure the duration of all incoming data bursts (unscheduled and scheduled) destined to egress link (j, k), over some fixed control interval, Δ. If the measured load on the egress link is greater than some predefined load threshold, ρ_{TH}, then a flow-rate reduction request (FRR) will be generated. We refer to such a link as being *congested*. The value of FRR, represented by $R_{j,k}$, explicitly indicates the *percentage* by which edge nodes must reduce the transmission rate of all burst flows (or LSPs) sharing link (j, k) in the next immediate control interval, $\Delta + 1$, and it is equivalent to

$$R_{j,k} = (\rho_{j,k} - \rho_{TH})/\rho_{j,k};\ \rho_{j,k} \geq 0. \tag{5.1}$$

When $R_{j,k}$ is set to zero, it indicates that no further change of transmission rate must be allowed. Whereas, $R_{j,k} = -1$ indicates that the source can increase its rate of transmission.

When a switch node n is overloaded ($\rho_{j,k} \geq \rho_{TH}$), the node will send a reduction request, $R_{j,k} > 0$, and no new FRR for link (j, k) will be sent until $\Delta + RTT$ time units later. This is to ensure that the change has actually taken place. However, the actual control interval in which the switch measures the average load is limited to one control interval, Δ. Therefore, as long as $R_{j,k} \leq 0$, FRR will be sent once every Δ time units. For simplicity and without loss of generality, we assume the value Δ is RTT, where RTT is the largest round-trip delay in the networks.

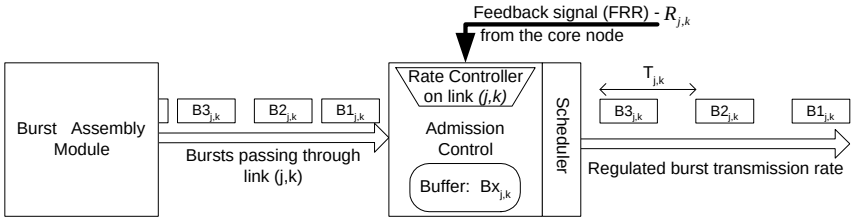


Figure 5.3. Flow control at the edge node using the proportional control algorithm with explicit reduction request (PCwER) scheme. The data burst transmission rate is adjusted by changing the data burst interdeparture time $T_{j,k}^{\Delta} = 1/\phi_{j,k}^{\Delta}$

5.4.2 Rate Controller Mechanism

The basic rate adjustment mechanism in PCwER is as follows. Upon receiving a negative FRR, the source will increase its rate of transmission of data bursts on link (j,k), $\phi_{j,k}^{\Delta-1}$, within the next control interval Δ by some fixed unit IR:

$$\phi_{j,k}^{\Delta} = \phi_{j,k}^{\Delta-1} + IR. \tag{5.2}$$

On the other hand, if $R_{j,k} > 0$, then the sending rate decreases as follows:

$$\phi_{j,k}^{\Delta} = \phi_{j,k}^{\Delta-1}(1 - R_{j,k}). \tag{5.3}$$

In the above expressions IR is constant and is called the *rate increase increment*. If $R_{j,k} = 0$, then $\phi_{j,k}^{\Delta-1} = \phi_{j,k}^{\Delta}$. It is evident that $R_{j,k}$ is a function of time and changes for each control interval Δ.

In practice, the data burst transmission rate is adjusted by changing the data burst interdeparture time $T_{j,k}^{\Delta} = 1/\phi_{j,k}^{\Delta}$, as shown in Fig. 5.3. In other words, every time the source sends a burst on link (j,k) it sets a timer value with a timeout equal to the inverse of the required transmission rate, and it transmits the next burst traveling on the same link when the timer expires. If the length of the previous bursts is such that the channel is busy at $T_{j,k}^{\Delta}$, then the new burst will be transmitted at the first available time instance, and the timer will reset. We will elaborate on burst scheduling in details in this section.

5.4.3 Rate adjustment algorithm

Upon receiving multiple FRR messages from different links, the edge node determines the most congested link (j, k) along each source-destination path and subjects all data bursts (or LSPs) passing through the congested link to a rate adjustment according to the increase/decrease functions described above. It must be noted that FRR signals are in fact asynchronous and can be received at different time intervals. However, as mentioned before, the requested rate change goes into affect only at the start of next transmission period, which is equivalent to $\Delta + RTT$. In our protocol we assume that each edge node keeps track of the latest values of the following parameters: average transmission rate along each link, $\phi_{j,k}$, the latest reduction request, $R_{j,k}$.

We now describe details of our proposed rate-based control algorithm. Upon receiving the reduction rate request on link (j, k) at time $t1$, $R^{t1}_{j,k}$, the edge node n takes the following actions:

Step 0: If multiple FRR signals are received for a path $\mathbf{R}(\mathbf{s}, \mathbf{d})$, find the FRR with the largest value: $R^{t1}_{j,k} = max\{R^{t1}_{j,k}, R^{t1}_{m,n}, ...\}$ for all $(m, n), (j, k), ... \in \mathbf{R}(\mathbf{s}, \mathbf{d})$.

Step 1: If $R^{t1}_{j,k} = 0$, continue transmitting at the current rate.

Step 2: If $R^{t1}_{j,k} < 0$, increase the transmission rate of all (s, d) flows where $(j, k) \in \mathbf{R}(\mathbf{s}, \mathbf{d})$, according to the increase function *iff* the following conditions satisfy:

- The most congested link on $\mathbf{R}(\mathbf{s}, \mathbf{d})$ was $(j.k)$ and $\phi_{s,d} + IR \leq \phi_{m,n}(1 - R_{m,n})$, where (m, n) is the next most congested link on $\mathbf{R}(\mathbf{s}, \mathbf{d})$. Another words, the rate increase should not cause congestion on any other link $(m, n) \in \mathbf{R}(\mathbf{s}, \mathbf{d})$.
- The rate increase cannot exceed the predefined link load threshold: $\phi_{j,k} + IR < \rho_{TH} \cdot (S \cdot W)$.

Step 3: If $R^{t1}_{j,k} > 0$, check the value and time of the last FRR message, $R^{t0}_{j,k}$ and $t0$, respectively

- If $t1 < \Delta + RTT + t0$, ignore the incoming $R_{j,k}^{t1}$.
- If $t1 > \Delta + RTT + t0$, reduce the transmission rate of all (s, d) such that $\mathbf{R}(\mathbf{s}, \mathbf{d})$ include (j, k): $\phi_{s,d}^{t1} = \phi_{j,k}^{t0}[1 - R_{j,k}^{t1}]$ where $(j, k) \in \mathbf{R}(\mathbf{s}, \mathbf{d})$.

Clearly, every time a new FRR is received and the rate of transmission is changed, all records, including the latest FRR value an the time of the latest rate change on each link, must be updated accordingly.

We illustrate the above concepts using the example shown in Fig. 5.4, where nodes S1, S2, and S3 are sending data bursts to Node S5 and Node S4 is the bottleneck. We assume the system is at equilibrium and rate of transmission is constant. As shown in the timing diagram in Fig. 5.4, S1-S3 send their data bursts at different instances, namely $t1$, $t2$, and $t6$. At time $t3$, the end of the first control interval, S4 detects congestion on the link between S4 and S5 and requests a reduction of 22%. Once S1 and S2 receive the new FRR, they reduce their sending rate accordingly. Note that the FRR signal which was initiated at $t3$ does not take into account the increased load caused by Node S3 at $t6$. The average value measured at $t9$, which is one control interval later ($t3 + RTT = t9$), is ignored by S4. At $t9$ a new averaging starts, and one RTT later ($t13$), another FRR signal is generated indicating the latest average measured load and sent back to S1-S3. After $\Delta + RTT$, ($t13 + \Delta + RTT$) all nodes are expected to reduce their transmission rate to meet the target load value requested by S4. At that time, the FRR is expected to be set to zero, indicating no further change in transmission rate is required. Note that after receiving and implementing the first FRR, source nodes ignore any other reduction request that involves the link between S4 and S5 until $\Delta + RTT$ later. For example, once the FRR is received at $t7$ and the transmission rate is changed, no other changes will be implemented until ($t14 = t7 + \Delta + RTT$).

When a link is congested and a flow-rate reduction request is sent to sources, if no new flows are added, it can take as long as $\Delta + RTT$ time units before the congestion is cleared out. Obviously, as the bandwidth-delay product increases more bursts will be subject to drop and more resources will be wasted before a congestion condition is resolved. Therefore, an alternative to *quickly* reduce injecting excessive bursts to downstream congested

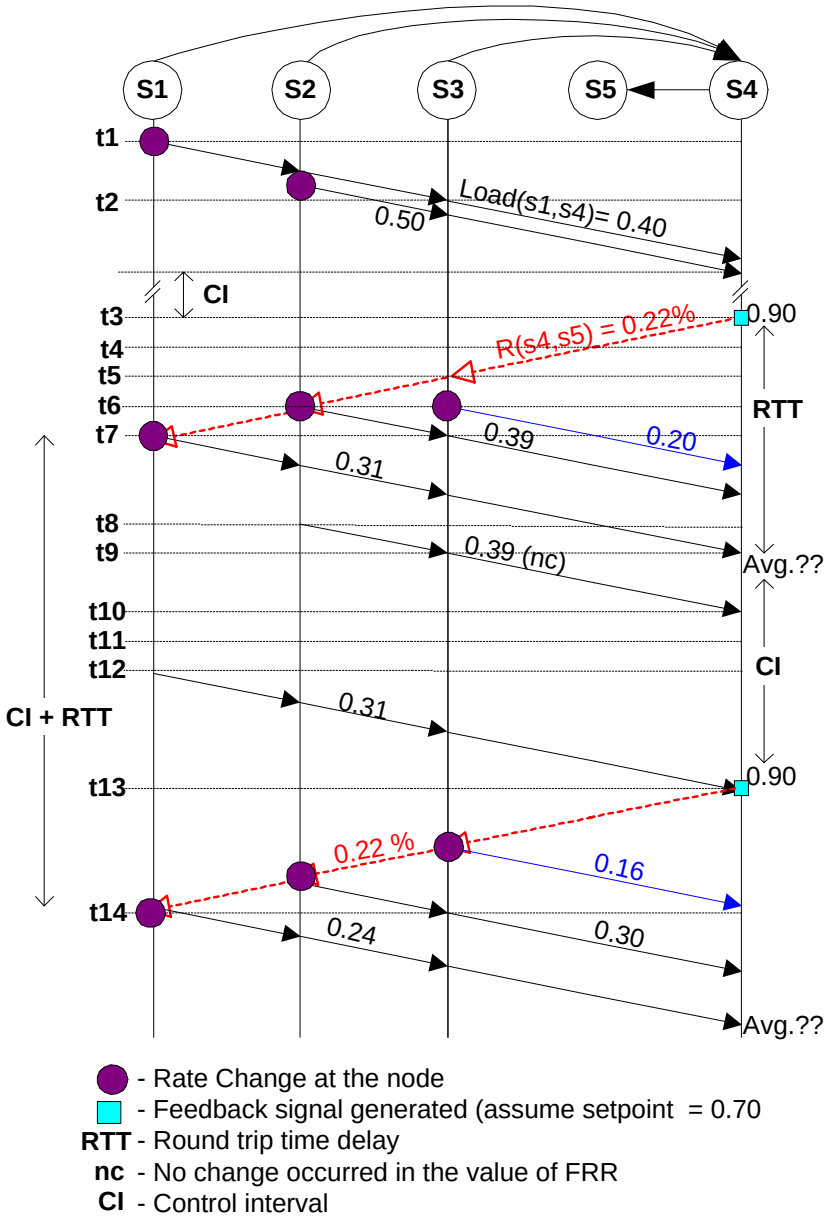


Figure 5.4. An example of a 5-node network and its feedback timing diagram. Control interval is equivalent to Δ.

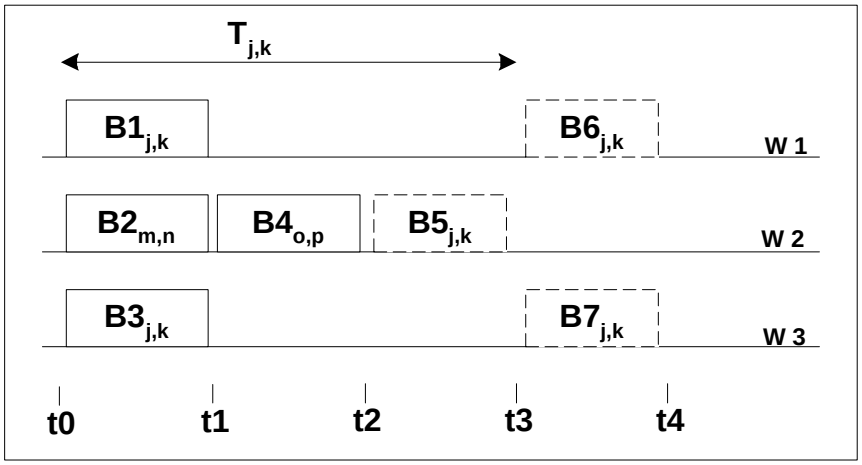


Figure 5.5. Illustrating the PCwER's scheduler operation in the edge node when the new data bursts, $B5$, $B6$, and $B7$ passing through link (j, k) arrive at $t1$, $t1$, and $t2$, respectively (arrival times are not shown in the figure). New bursts $B5$, $B6$, and $B7$ are delayed and scheduled on channels $W1 - W3$.

links is to ask upstream nodes to temporarily drop the bursts which will be passing through the congested link. Such *intentional* burst dropping at upstream nodes will continue until source nodes reduce their transmission rate of burst flows. This scheme, called *PCwER with intentional dropping (PCwER-ID)*, appears to be resource efficient in the sense that no resources are wasted by bursts attempting to pass through the congested link. A clear disadvantage of this approach is that the feedback messaging must be processed by all upstream nodes.

5.4.4 Scheduler

Data bursts subject to admission control must be scheduled on available wavelengths (channels). The admission control's scheduler, as shown in Fig. 5.3, performs as follows. An admitted burst, $Bx_{j,k}$, will be scheduled on the latest available wavelength where the interarrival time between burst x passing through link (j, k) and the last scheduled burst, y, on the same wavelength is at least equal $T_{j,k}$ time units. If no such wavelength exists, the burst must be further delayed until some units of time later. Clearly, if a burst arrives after $T_{j,k}$ time units, it will be conforming and thus the counters are reset and the burst will immediately be transmitted.

These concepts are shown in Fig. 5.5. In this example data bursts $B1$, $B2$, $B3$, and $B4$ are already scheduled and new bursts $B5$, $B6$, and $B7$ arrive at times $t1$, $t1$, and $t2$, respectively (arrival times are not shown in the figure). All new bursts will be passing through link (j, k) and we assume the minimum interdeparture time for link (j, k) is $T_{j,k}$. The latest available channel for $B5$ to be scheduled is channel 1 (or 3). $B6$ and $B7$ cannot be scheduled before $t3$. Thus, channel 1 (or 3) can be used to schedule either $B6$ or $B7$ after delaying their transmission for 2 and 1 time units, respectively. Note that $B5$ cannot be scheduled before $t3$ on channel 1 (or 3). Thus, $B5$ will be delayed by one time unit and scheduled on channel 2 at $t2$.

5.5 Analysis

In order to analyze our proposed rate-based contention control model for OBS network, we consider a continuous-space deterministic (or fluid) model [136] as shown in Fig. 5.6(a). We use this model to address four important issues: (1) to determine how fast the transmission rate should increase when the system is underloaded; (2) to find the worst case instantaneous probability of loss at equilibrium when the desired system load setpoint, ρ_{TH} is given; (3) to find the convergence time for the system to approaches; (4) and to determine how fairly the bandwidth is distributed between different competing sources. However, before we address these issues, we look at the impact of various design parameters.

5.5.1 Design Parameters

The admission control mechanisms can be very sensitive to the parameter settings. In this section we evaluate the importance and affects of some of the design parameters.

(a) *Control interval*, Δ: The average elapsed time for the FRR to reach an edge node is proportional to the network diameter times the average transmission delay on each link. The transmission delay is defined as the time it takes a signal to travel between two adjacent nodes. Obviously, as this elapsed time increases, the network becomes less responsive to load changes. In general, if the value of Δ is too small, the number of feedback signals

will increase. On the other hand, if Δ is too large, the feedback mechanism will be insensitive to the moderate changes in network load. Therefore, various factors including the network topology, traffic characteristic, and average transmission delay can be considered in determining the value of Δ. In this section we assume $\Delta = 2 \cdot RTT$.

(b) *Switch load threshold, ρ_{TH}*: The value of the switch load threshold, ρ_{TH}, also impacts the system performance. If the value of ρ_{TH} is too high, the admission control becomes less effective. On the other hand, if ρ_{TH} is very small, the admission control will be activated too quickly. This in turn, results in generating higher number of feedback messages, thereby increasing the control overhead in the network and leading to a potentially instable system. Furthermore, if ρ_{TH} is too low, the network can be unreasonably under-utilized.

(c) *Oscillation of load around the setpoint: $\rho_{TH} \pm \epsilon$*: Assuming that the measured traffic load on a link is around ρ_{TH}, any small changes in the offered load by the source on that link can result in FRR oscillation. One way to prevent this is by setting a lower and upper threshold such that $\rho_L = \rho_{TH} - \varepsilon$ and $\rho_H = \rho_{TH} + \varepsilon$, where ε is a small percentage of ρ_{TH}. Hence, the source will not be permitted to change its traffic load to the near-congested link unless the measured load drops below ρ_L or rises above ρ_H.

(d) *Calculating the value of FRR, $R_{j,k}$*: Accurate computation of $R_{j,k}$ results in fast convergence and reduction of data burst flow-rate on a congested link and hence, lowering the data burst loss. On the other hand, it is critical not to under-utilize the network. As indicated by Eqn. (5.1), the FRR value is calculated as a function of the measure load. However, there are different methods in which the load, $\rho_{j,k}$, can be measured on each egress port. In the following paragraphs we describe three approaches to measure $\rho_{j,k}$ in Eqn. (5.1).

(a) *Measuring the carried load (MCL)*: The rate reduction request can simply represent the carried load on an output link of the switch within the previous control interval, $\Delta - 1$, $\rho^{\Delta-1}(P_{eg})$. In this case the $R_{j,k}$ does not include the number of unscheduled

(or discarded) data bursts:

$$\rho_{j,k} = \rho_{j,k}(P_{eg}). \tag{5.4}$$

One disadvantage of this approach is that in order to reduce the data burst flow on the overloaded link, it may be necessary to send several FRRs. Consequently, load reduction occurs slowly and more bursts are expected to be lost until the overload condition is resolved. This becomes more critical as the bandwidth-delay product becomes more significant.

(b) *Measuring the total load (MTL)*: Another approach to calculate $\rho_{j,k}$ is to compute the total incoming load on all ingress ports destined to the output link of the switch:

$$\rho_{j,k} = \Sigma_{i \in P_{in}} \rho_{j,k}(i) = \rho_{j,k}(P_{eg}) \cdot (1 + P_B(P_{eg})). \tag{5.5}$$

Note that as shown in the above relation, measuring the total input load is equivalent to the sum of the carried load, $\rho_{j,k}(P_{eg})$, and the ratio of bursts blocked, $P_B(P_{eg})$, on the egress port P_{eg}. Thus, the explicit reduction rate is calculated based on the overall load, including all scheduled and unscheduled data bursts destined to each link, and sent back to edge nodes. A major advantage of this scheme is its fast convergence property. That is, after the first FRR, we can expect the edge nodes to properly respond to the flow reduction request and reduce their load on the congested link, assuming there has been no changes in the network. However, the basic drawback of this approach is the need for larger counters monitoring each egress port and thus, higher hardware requirements. In our protocol we consider this case.

(c) *Measuring the expected load (MEL)*: Assuming upstream nodes do not misbehave and properly control their carried load on each of their egress ports such that $\rho_{j,k}(P_{eg}) \leq \rho_{TH}$, a trivial improvement to PCwER is to calculate the input load based on the expected load value in the next control interval, $\Delta + 1$. This is based on the assumption that if $\rho_{j,k}(P_{in}) > \rho_{TH}$, the upstream node has already sent an FRR request to

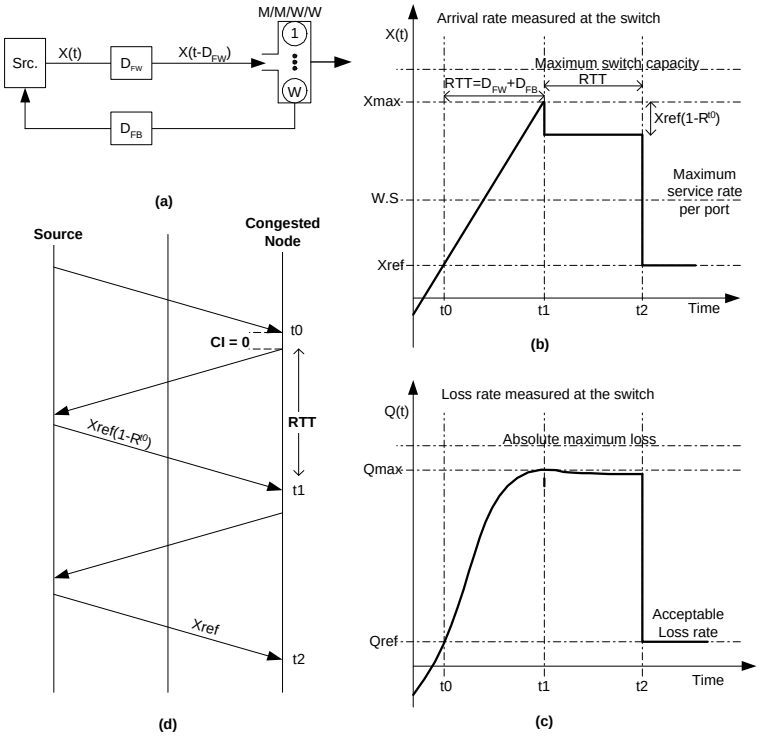


Figure 5.6. (a) A continuous model of the PCwER contention control system; (b) data burst arrival measured at the bottleneck node; (c) corresponding data burst loss rate.The control interval is ignored in the figure; (d) timing diagram of the feedback signal.

appropriate source nodes to reduce their load. Consequently, $\rho_{j,k}$ can be calculated using

$$\rho_{j,k} = \Sigma_{i \in P_{in}} min(\rho_{j,k}(i), \rho_{TH}). \tag{5.6}$$

Similar to case (b), a clear disadvantage of this approach is the need for added hardware in order to measure the incoming load on each ingress port.

5.5.2 Algorithm Convergence

In our model, as shown in Fig. 5.6(a), we assume the bottleneck is a multi-server bufferless queueing system with W individual servers, each with a service rate of S bursts/second, and we let $X(t)$ denote the aggregated transmission rates from all source nodes in bursts/second at time t. Furthermore, we assume D_{FW} and D_{FB} represent the propagation delays from the source to the bottleneck node with the congested link and from the bottleneck node to

the source, respectively. Due to the bufferless nature of our system, no queueing delay is applied to our model.

Under continuous-space assumption, we can redefine the rate increase/decrease algorithm as follows:

$$X(t) = \begin{cases} X(t_0) + a \cdot (t - t_0) & \text{if increasing} \\ X(t_0) \cdot b(t - t_0) & \text{if decreasing;} \end{cases} \tag{5.7}$$

where t is the current time and t_0 is the time the FRR signal is sent to the source. Relating the above continuous expressions and Eqn. (5.2)-(5.3) we allow $a = IR \cdot \Delta$ and b to be a continuous decreasing function equivalent to $1 - R_{j,k}$. Since we are interested in the worst case loss rate, the actual function of $b(t)$ is not critical. In this section we only consider the equilibrium condition when no new flow of bursts is added and no active flow is terminated. In addition, we assume that there is only one congested link.

The behavior of $X(t)$ as a function of time and the corresponding loss rate are shown in Fig. 5.6(b)-(d). The maximum loss will occur when the transmission rate reaches its maximum level at t_1, as shown in Fig. 5.6(b). Hence, we are interested to find X_{max}. From Fig. 5.6(b), it is clear that the elapsed time between when the feedback signal (FRR) is generated and the time the aggregated transmission rate reaches its maximum level is $D_{FW} + D_{FB}$. The propagation delays between all nodes are considered to be the same, $RTT = D_{FW} + D_{FB}$. Note that the control interval is ignored in this case and we assume FRR is generated as soon as the measured load increases beyond the setpoint. Therefore, the maximum arrival rate received by the bottleneck node will be

$$X_{max} = X_{ref} + a \cdot RTT. \tag{5.8}$$

Consequently, the maximum experienced load at equilibrium state with link fluctuation around ρ_{TH}, will be $\rho_{max} = (X_{max}/S \cdot W)$. Note that when the FRR with value of $R_{j,k}^{t0}$ is received at $t1$, the new rate will be $X_{ref} \cdot (1 - R_{j,k}^{t0})$.

Using the well-known Erlang-B formula, the burst loss probability can be calculated as

$$P_{loss}(\rho) = \frac{\rho^W / W!}{\Sigma_{k=0}^{W} \rho^k / k!}. \tag{5.9}$$

Consequently, if the arrival rate is X_{max}, the maximum burst loss rate (in bursts/second) can be expressed as $Q_{max} = P_{loss}(\rho_{max}) \cdot X_{max}$. The percentage difference of the maximum loss rate from its target value can be expressed as $\delta_{ref} = (Q_{max} - Q_{ref})/Q_{max}$, where $Q_{ref} = P_{loss}(\rho_{TH}) \cdot X_{ref}$ and $X_{ref} = \rho_{TH} \cdot (S \cdot W)$.

Using the above relationships, it can be seen that, given the target loss rate and its maximum acceptable instantaneous fluctuation, Q_{ref} and δ_{ref}, respectively, we can determine the values of ρ_{TH} and IR.

As shown in Eqn. (5.8), the maximum loss rate is tightly related to IR, and the round-trip delay. Larger values of IR result in faster convergence to the target link load, and hence, higher throughput. The trade off, however, is a higher maximum loss rate. A closer look at Fig. 5.6(b)-(d) shows that, under the equilibrium condition, the system approaches the link load threshold, ρ_{TH} is at least $\Delta + RTT$ time units.

5.5.3 Algorithm Fairness

Fairness is considered to be an important issue in any rate-based contention avoidance network with feedback. A widely adopted criterion to define fairness is knows as *maximum fairness* criterion. In this scheme, the traffic flows from different edge nodes with the same priority must have an equal share of the congested link, $S \cdot W/|N|$. Such a property is quantified by a *fairness index* defined as follow:

$$FI(X) = \frac{(\Sigma X_i)^2}{|N|(\Sigma X_i^2)}, \tag{5.10}$$

where $|N|$ is the number of concurrent flows into the congested link and X_i is the sending rate of the ith flow at equilibrium. Typically, FI is a value between 0 and 1 with $FI = 1$ indicating perfect fairness. Based on this definition, it can be seen that using PCwER when there is no congestion in the network and transmission rate is increasing linearly, FI tends to increase and approach unity: $1 \geq FI(X + \alpha) \geq FI(X)$. On the other hand, when link (j, k) is congested and edge nodes must decrease their sending rate, the value of FI does

not change:

$$FI(X - X \cdot R_{j,k}) = FI(X). \tag{5.11}$$

These results indicate that our proposed rate-based adjustment algorithm stabilizes the total load around the desired target value (ρ_{TH}) and it does not change the initial ratio of offered loads by different competing edge nodes in the OBS network due to congestion.

Assuming that the network is in equilibrium and the rate of incoming traffic into one of the sources increases, the node can increase its rate at some higher rate than its original rate, $X_{org} + \theta$. Hence, the congested link will experience an increase in its average load by θ. Consequently, the bottleneck node must send a new FRR requesting all nodes to reduce their rate by $\theta/(\rho_{TH} + \theta)$. Note that when a node is starved, typically its transmission rate (X_{org}) is very small and the measured load on the switch is very close to ρ_{TH} leading to FRR value to be zero. This permits the node with excess traffic to increases its transmission rate from X_{org} to

$$(X_{org} + \theta) \cdot \frac{\rho_{TH}}{\rho_{TH} + \theta}. \tag{5.12}$$

One approach to control the relative contributed load on the congested link by each of the sources is to measure each ingress edge node's contribution to the congestion link. In this case, a core node must maintain $|N| - 1$ sets of information for each of its $|P_{eg}|$ egress ports, where $|N|$ is the number of edge nodes in the network. This will require as many as $|P_{eg}| \cdot (|N| - 1)$ individual counters in the switch and considerable increase in the number of feedback messaging communicated between the nodes. The intuitive trade-off of this complexity is, however, achieving a better resource allocation among all edge nodes and protecting well-behaved edge nodes from malicious ones.

5.6 Performance results

In this section we discuss the simulation results obtained by implementing the proposed data burst admission control in a feedback-based OBS network. We consider the NSFnet

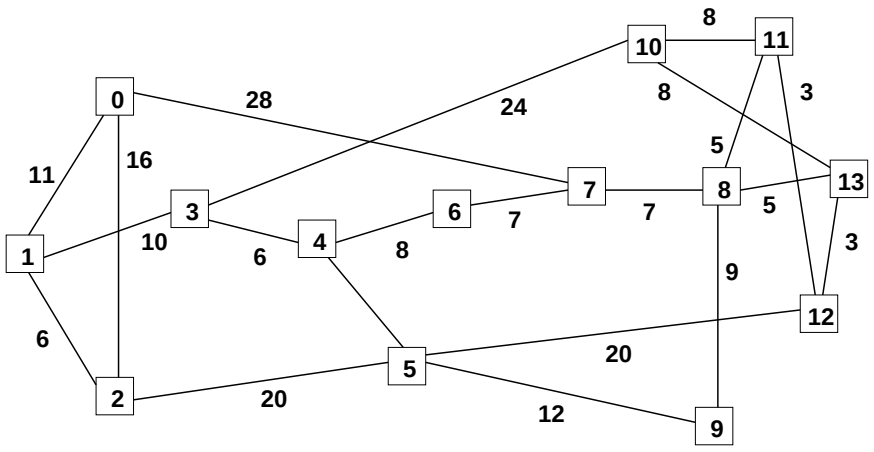


Figure 5.7. The NSF network with 14 nodes and 21 bidirectional links.

backbone, shown in Fig. 5.7, as our test network. Six ingress/egress node pairs are selected to carry active traffic, namely (13,11), (4,8), (6,11), (0,13), (9,11). The bottleneck links are (6,18) and (7,8). In this network, we assume the RTT delay between each node pairs is different and is between 10-50 ms. We also consider the following assumptions for the simulation environment: burst length is fixed and is equivalent to 100 μs, containing 1250 bytes; the transmission rate is 10 Gbps with 4 wavelengths on each link; the switching time is 10 μs; and the burst header processing time at each node is assumed to be 2.5 μs. Furthermore, we assume full wavelength conversion at every node and adopt the latest available unscheduled channel (LAUC) algorithm to schedule data bursts at the core nodes. The design parameters for the PCwER congestion avoidance mechanism are as follows: $IR = 0.075$, $\rho_{TH} = 0.7$ and ρ_L and ρ_H are 0.695 and 0.705 respectively. Unless otherwise stated, we assume FRR values are measured using MTL approach according to Eqn. (5.5).

In our C-based simulation model we used confidence interval accuracy as the controlling factor. For each case of interest, the simulation was run until a confidence interval level of 90% was observed and an acceptably tight confidence interval (5%) were achieved. Calculations of the confidence interval were based on the variance within the collected observations [108]. All simulations were performed on a UNIX-based multiprocessor machine.

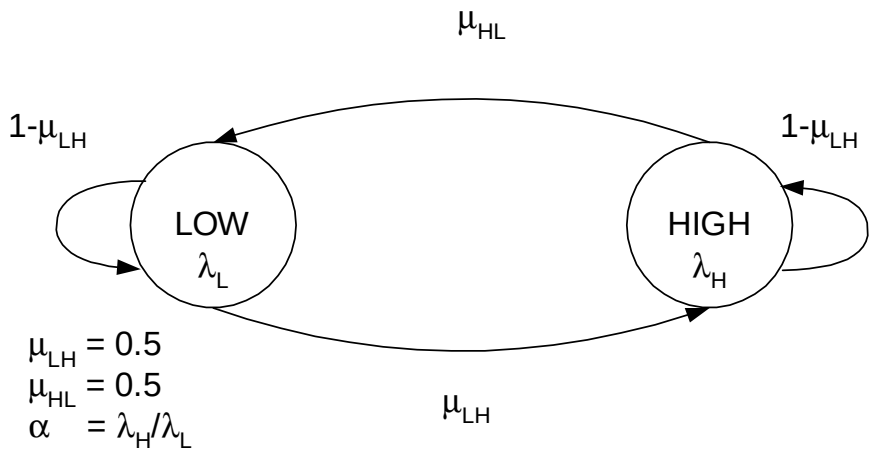


Figure 5.8. Two-state Markov modulated arrival process.

We represent the simulation results in terms of network load. We define burst blocking probability as the percentage of IP packets that are sent transmitted by the edge node source but never received.

5.6.1 Traffic model

The traffic model we consider in our study is characterized by two random processes modeling both the spatial and the temporal characteristics of the arriving data bursts. The spatial characteristic, which indicates the distribution of data bursts destinations is modeled by a uniform distribution. On the other hand, the process of modeling the inter-arrival times between successive data burst arrivals is based on a two-state Markov chain, as shown in Fig. 5.8, consisting of a HIGH and LOW state. In the HIGH state, assembled data bursts arrive at rate λ_H , which is higher than the average arrival rate λ_{avg}. In the LOW state, fewer IP packets arrive and thus burst arrival occurs at $\lambda_L < \lambda_{avg}$.

In each state we consider exponentially distributed burst inter-arrival times. Similarly, the time that the system remains in each state is exponentially distributed. The average data burst arrival rate in this model is determined by $\lambda_{avg} = \lambda_H \cdot \mu_H + \lambda_L \cdot \mu_L$, where the state probabilities μ_L and μ_H are computed as $\mu_H = \frac{\mu_{HL}}{\mu_{HL}+\mu_{LH}}$ and $\mu_L = \frac{\mu_{LH}}{\mu_{HL}+\mu_{LH}}$. Thus, the

average data burst arrival rate will be

$$\lambda_{avg} = \lambda_H \cdot \frac{\mu_{LH}}{\mu_{HL} + \mu_{LH}} + \lambda_L \cdot \frac{\mu_{HL}}{\mu_{HL} + \mu_{LH}}. \tag{5.13}$$

Three possible scenarios can be considered:

1. $\lambda_L = \lambda_H$; In this case the model is reduced to an exponential arrival with fixed size data bursts.

2. $1 > \lambda_H > \lambda_L > 0$; In this case the arrival rate varies between λ_H and λ_L as the time increases. We refer to as the traffic persistency factor. We define $\alpha = \lambda_H/\lambda_L$ as the *traffic persistency factor*. Note that if $\alpha = 1$, we obtain a Poisson arrival model, and as α increases, the traffic becomes more bursty.

3. $\lambda_H = 1$ and $\lambda_L = 0$; This case represents an ON-OFF bursty traffic model in which bursts of traffic arrive in the state HIGH (ON). No traffic is generated in the LOW (OFF) state.

In this study we only focus on cases (1) and (2).

5.6.2 Simulation results

Fig. 5.9 shows the probability of burst loss for Poisson arrivals. This figure compares the probability of data burst loss with and without the PCwER congestion avoidance mechanism. As the load threshold in the switch drops, the loss probability decreases. This occurs as a result of *choking* the source and lowering the maximum transmission rate allowed on the bottleneck link. When the total load at the bottleneck link reaches ρ_{TH}, a slight increase in loss rate is experienced, which is due to the RTT. The trade-off of reducing the overall loss rate due to rate-control is lowering the link throughput, as shown in Fig. 5.10. This figure shows the normalized throughput for an exponentially distributed traffic model with and without the contention avoidance mechanism. The maximum achievable data link capacity in our model is 40Gbps. The value of the load threshold of the switch directly

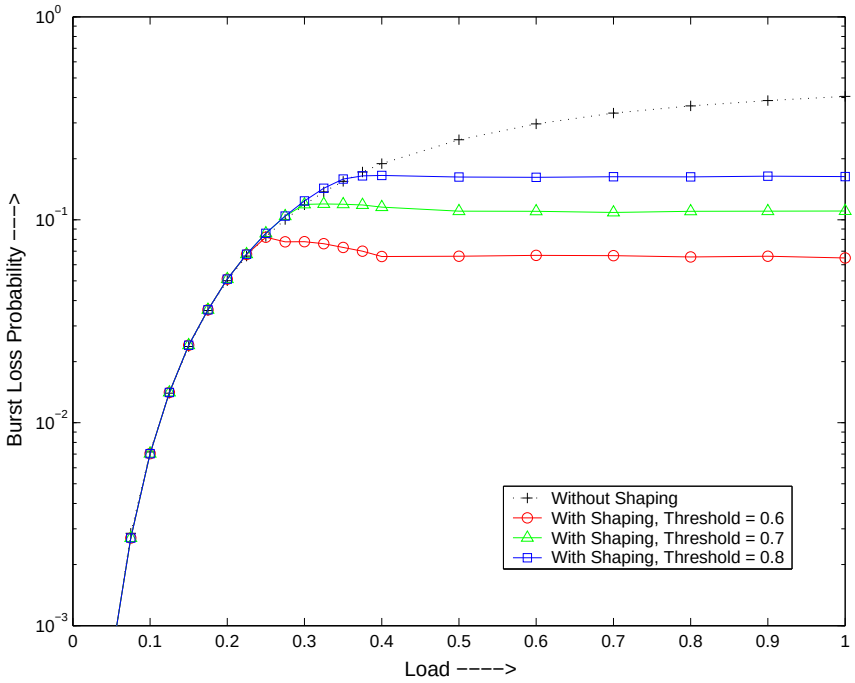


Figure 5.9. Comparing the probability of data burst loss with and without contention avoidance when traffic is exponentially arriving for different values of ρ_{TH}: 0.6, 0.7, and 0.8.

impacts the network throughput. For example, as shown in Fig. 5.10, when the threshold is set to 0.7, the throughput of the bottleneck link at high loads will be 0.59 (40 Gbps) = 23.6 Gbps, compared to 0.81 (40 Gbps) = 32.4 Gbps when no contention avoidance is implemented. However, the loss at ρ_{TH} = 0.7 is significantly lower. Note that as long as the measured load remains above the threshold, the system stays in continuous choking state. Similar results in terms of data burst loss probability and link throughput can be observed when the traffic is Poisson arriving with high and low arrivals, as shown in Fig. 5.11 and Fig. 5.12. When the threshold value is low, such as 0.6, as the measured load on the bottleneck reaches the threshold, the probability of loss continues to increase until the links are overloaded and the system goes into the choke state. The probably of loss in case of exponentially distributed traffic with high/low averages experiences more variations around the threshold level. This is because in order for the system to go into choke state higher average load is required.

Focusing on the Poisson traffic with high and low arrivals, we now examine the performance of PCwER with and without intentional dropping on intermediate nodes. Fig. 5.13

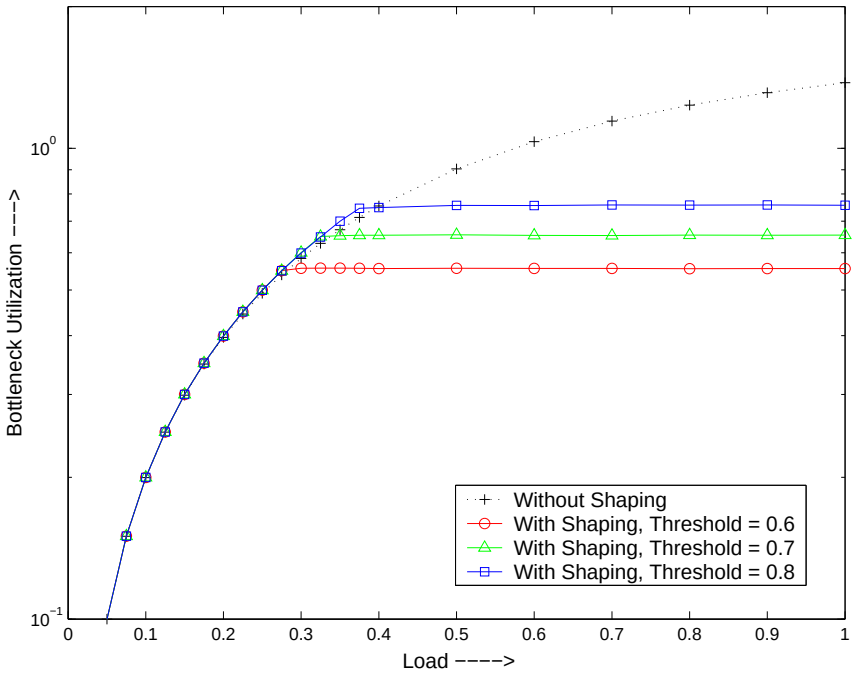


Figure 5.10. Normalized throughput when the traffic is exponentially arriving.

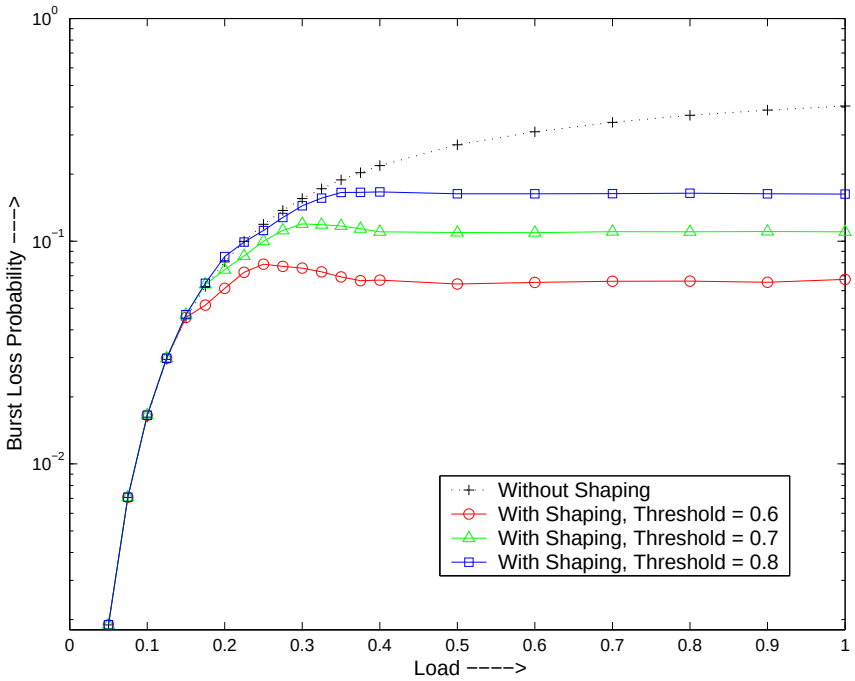


Figure 5.11. Comparing the probability of data burst loss with and without contention avoidance with variant rate traffic for different values of ρ_{TH} when the persistent factor is 3.

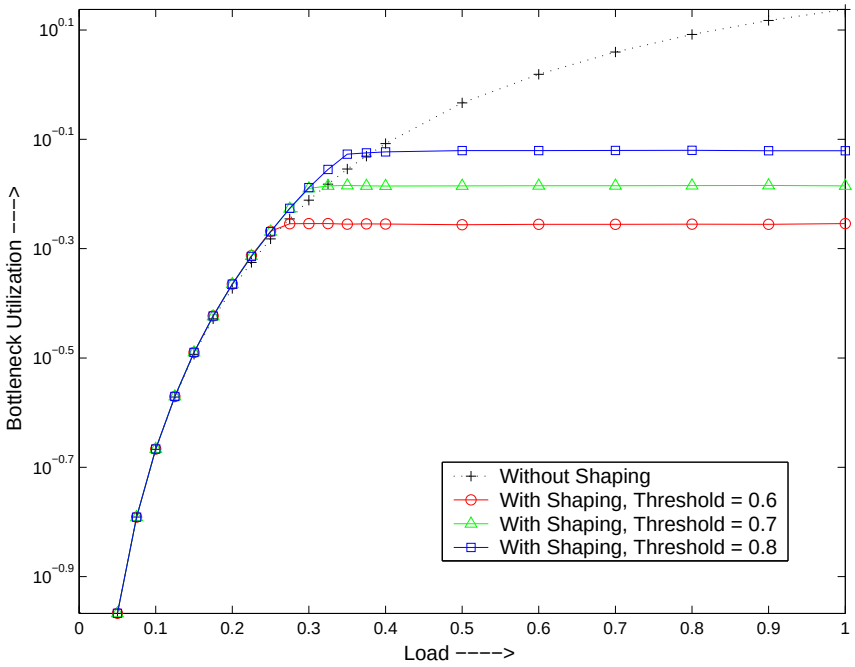


Figure 5.12. Normalized throughput when the persistent factor is 3 .

shows that, in general, the performance of PCwER with intentional dropping is better due to its ability to reduce burst transmission to links which are congested and thus providing better resource efficiency. This figure suggests that intentional dropping is more effective as the total round-trip delay in the network increases. This is due to the fact that larger RTT will result in longer congestion state and hence more resources will be waisted.

Next, we look at the performance of the PCwER when the FRR signal is calculated in different ways, namely, MCL, MTL, or MEL approach, as described in Section 5.5.1. Fig. 5.14 suggests that the best performance is achieved by MEL, where load is measured according to its expected value from the upstream adjacent nodes. Note that the main difference occurs when the network load is relatively high and less than the load threshold, $\rho_{TH} = 0.8$. The main reason that MEL approach performs better is contributed to the fact that it can clear out the congestion state faster and it is more resource efficient. In MCL approach, when a link is congested the carried load is measured and hence it may take at least $2 \cdot (\Delta + RTT)$ time units before the source reduces its load to the proper level.

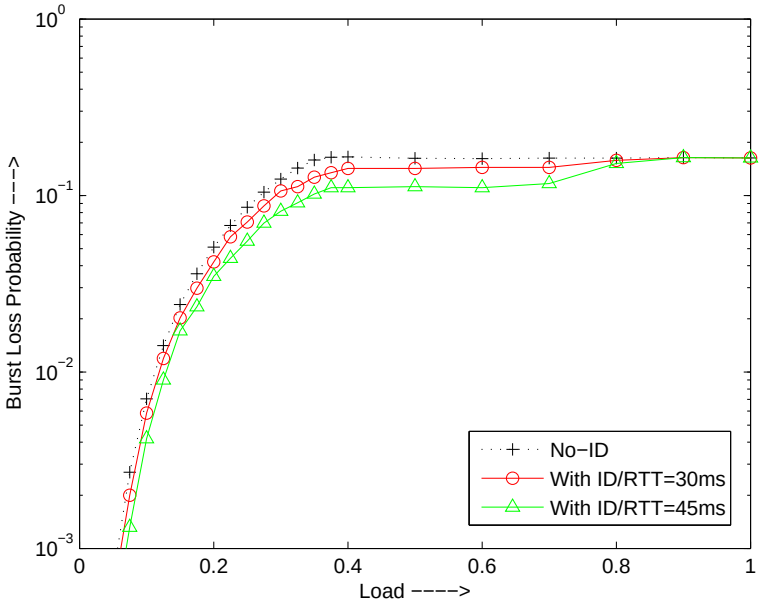


Figure 5.13. Comparing the burst loss probability in PCwER and PCwER-ID for different values of RTT. We assume $RTT = 30\ ms$ when PCwER with no intentional dropping is implemented.

Similarly, using MTL approach, adjacent nodes can potentially send redundant requests to the source, which results in longer persistent of congestion state.

The impact of rate increase increment (IR) is shown in Fig. 5.15-5.16. Note that as the value of IR increases, higher burst loss probability occurs. This is due to the fact that higher values of IR result in sharper increase of load and consequently higher link congestion. On the other hand, if the value of IR is small the load change occurs at much smaller rate and hence congestion detection occurs faster and fewer number of bursts will be lost. The downfall, however, is that as the value of IR reduces, the throughput decreases as well. This is shown in Fig. 5.16. In our experiment we have found that $IR = 0.075$ results in a good compromise between overall network throughput and loss.

The value of IR also impacts the maximum instantaneous burst loss. Fig. 5.17 shows that as IR increases the maximum loss on link (6,18) tends to become larger before the source edge nodes reduce their loads to the appropriate level. As we mentioned earlier, higher IR value indicates sharper ramp-up in load increase. In this figure, we assume

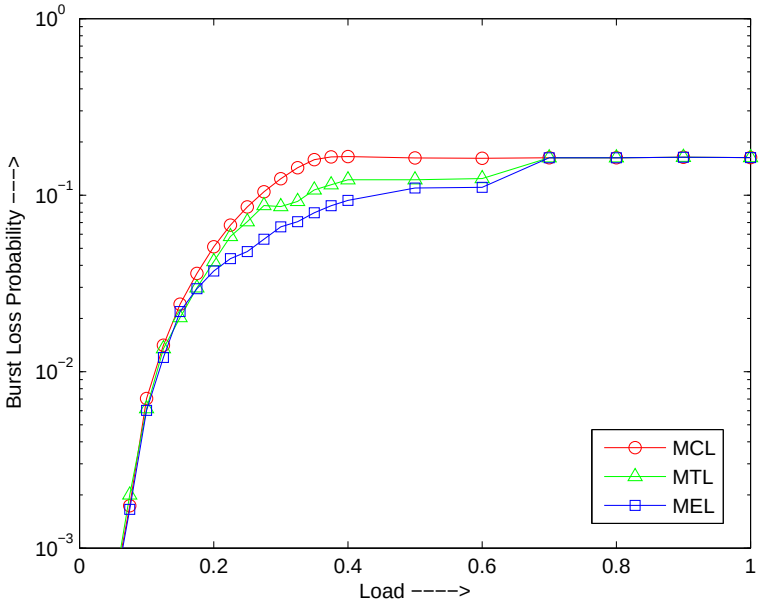


Figure 5.14. Comparing the burst loss probability in PCwER when the FRR signal is calculated based on MCL, MTL, and MEL approach. The load threshold is $\rho_{TH} = 0.8$.

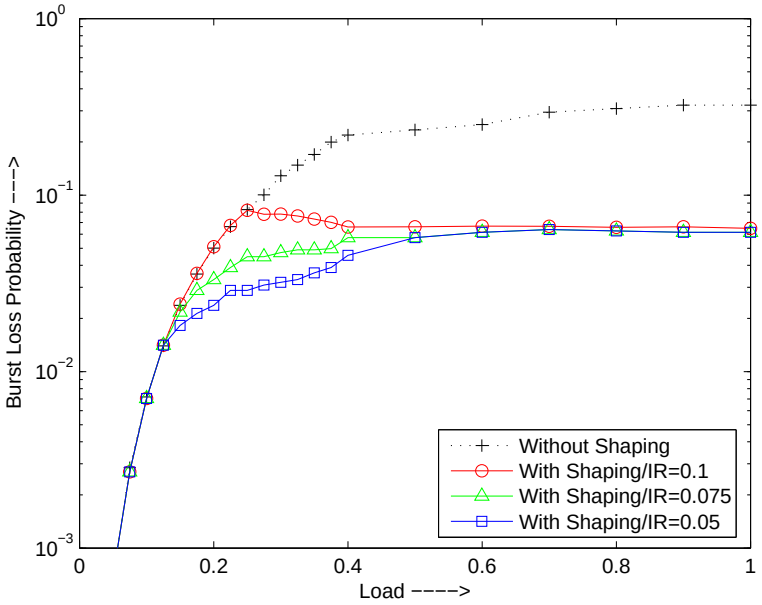


Figure 5.15. Burst blocking probability using the PCwER algorithm as IR changes between $\{0.100, 0.075, 0.050\}$.

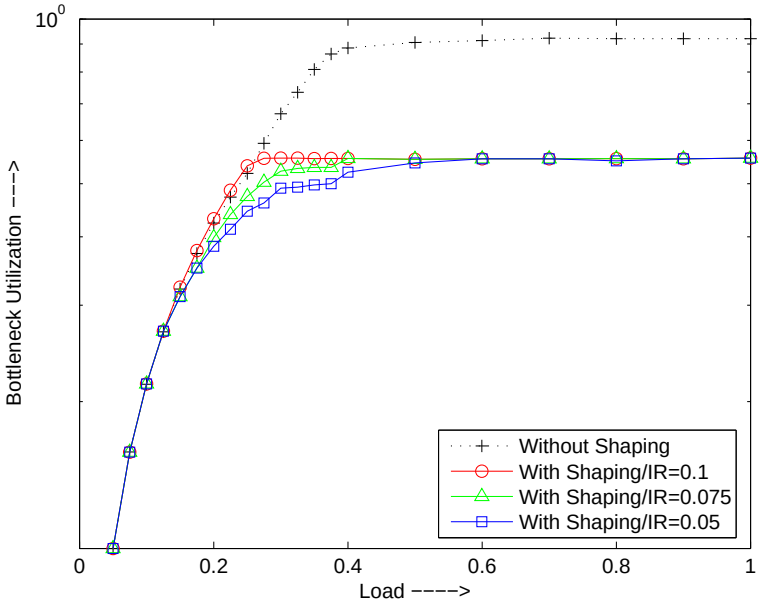


Figure 5.16. Bottleneck throughput using the PCwER algorithm as IR changes between $\{0.100, 0.075, 0.050\}$.

RTT is 30 ms, $\rho_{TH} = 80$, and nodes randomly change their transmission rate. Each measurement point in Fig. 5.17 is based on the average *blocking ratio* over $0.1 \cdot RTT$. We define average blocking ratio as the ratio of the number of bursts dropped over the total number of bursts destined for the egress port. Note that the time it takes for the load to settle to the setpoint value is about the same regardless of the value of IR. In this experiment we assumed using MEL approach.

We now consider a case in which bursts have a maximum end-to-end delay tolerance and cannot be delayed longer than T_{Max}. Therefore, for an incoming burst passing through the congested link (j, k), if $T_{j,k} > T_{Max}$ then the burst will have to be dropped at the source. We are interested in examining the impact of T_{Max} in burst loss probability. Fig. 5.18 shows the performance of PCwER as T_{Max} changes from 100 to 300 μs, indicating a single or three burst unit delay tolerance, respectively. We also compare these results with the case when T_{Max} is infinity, indicating boundless end-to-end delay tolerance. As the T_{Max} value decreases, the PCwER becomes less effective for moderate load values. Note that when the load is very high, PCwER performs almost the same for T_{Max} equal to 100 or 300 μs.

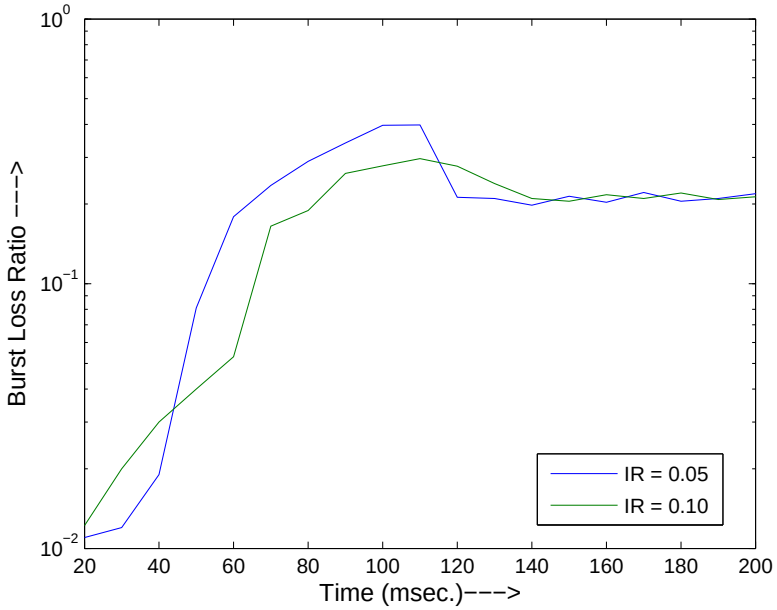


Figure 5.17. Burst loss ratio as a function time for different values of IR.

Similarly, when the network load is very low, only a few congestion cases occur. The results in Fig. 5.18 can also be verified by the fact that the maximum admission control delay, $T_{j,k}$, is bounded by $[0, 1 - \rho_{TH}]$. Hence, for very high loads (~ 1), $T_{j,k}$ is very limited. On the other hand, if the network load is very low, the PCwER algorithm is implemented only in rare cases.

5.7 Conclusion

In this chapter we proposed a rate-based contention avoidance mechanism for optical burst switching networks. Our proposed scheme, the proportional control algorithm with explicit reduction request (PCwER), significantly reduces the packet loss probability in the OBS network. The basic trade-off of PCwER is, however, the overall reduction of network utilization due to invoking admission control when the network is congested. Using a simple fluid model, we analyzed the characteristics of the control algorithm. Furthermore, through simulation, we compared the overall data burst loss with and without the PCwER contention avoidance mechanism. We showed that network throughput reduction, due to

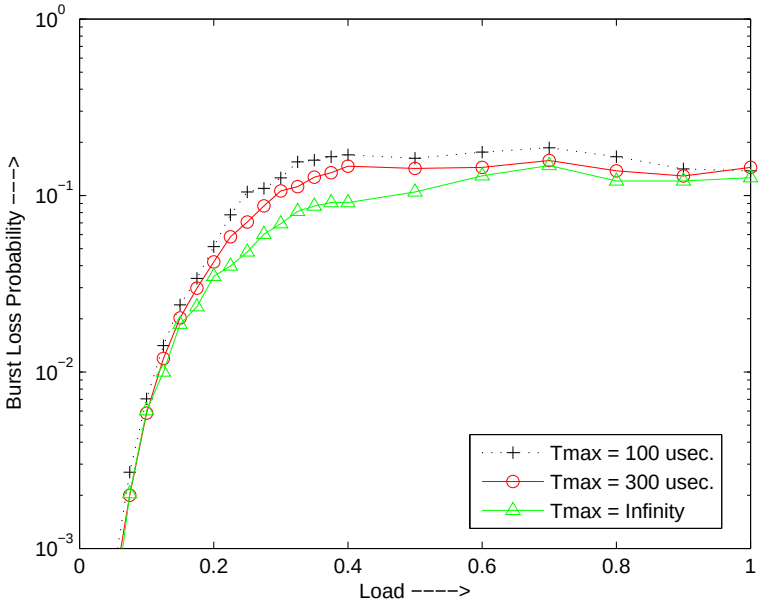


Figure 5.18. Burst loss probability as the maximum end-to-end delay tolerance, T_{max} changes.

rate control, is tolerable. An attractive feature of PCwER is that it can guarantee the worst case loss rate and hence can be used to support service differentiation.

One area of future work will be to extend the proposed PCwER framework such that it can support service differentiation and QoS. Another important issue which we did not examine in this chapter is to look at burst overlapping at the edge node and find a correlation between the burst rate reduction request and the overlapping factor.

CHAPTER 6
DYNAMIC TRAFFIC GROOMING IN OPTICAL BURST-SWITCHED NETWORKS

6.1 Introduction

In OBS networks, incoming data is assembled into basic units, referred to as *data bursts*, which are then transported over the optical core network. Control signaling is performed out-of-band by control packets which carry information such as the length, the edge node destination address, and the QoS requirements of the optical data burst. The control packet is separated from the data burst by an offset time, allowing the control packet to be processed at each intermediate node before the data burst arrives. Aggregating IP packets into large sized bursts can compensate for slow switching time at core nodes. This is motivated by the fact that the relatively mature MEMS-based optical crossconnects can provide a connection switching time of about 10 ms [137]; on the other hand, the typical switch reconfiguration time requirement for optical packets can be in order of microseconds (or even nanoseconds). Consequently, core nodes with slower switching times require larger *minimum burst lengths* in order to minimize the switching overhead.

An important issue in OBS networks is data burst assembly. Burst assembly is the process of aggregating IP packets with the same characteristics, such as edge node destination, class of service, etc., into a burst at the edge node. 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 when the time-out event is triggered. In a threshold-based approach, a limit is placed on the number of packets contained in each burst. A more efficient assembly scheme can be achieved by combining the timer-based and threshold-based approaches [48], [138], [49], [120].

IP packets assembled in a data burst have a time delay constraint, called *maximum end-to-end delay tolerance*, determining the deadline by which the packet must reach its OBS

destination. Thus, the main motivation for implementing the timer-based burst assembly approach is to ensure an IP packet doesn't wait at the edge node's assembly unit indefinitely before its maximum end-to-end delay tolerance is violated. If the arrival rate of incoming IP packets with the same characteristics is low, bursts are timed out and released before they reach their minimum burst length requirement determined by the core node switching time. Under such conditions, the timed out burst is smaller than the minimum length requirement. We refer to these short bursts as *sub-bursts*. Padding overhead must be added to sub-bursts in order to satisfy the minimum length requirement. However, excessive padding results in high link utilization and data burst blocking probability. Furthermore, when data bursts are timed-out, their aggregated IP packets will experience higher average delay. These concepts are illustrated in Fig. 6.1. In case (a) the data burst reaches its maximum size before it is timed out. Case (b) represents a situation in which the burst is timed out before it reaches its maximum size. In case (c) the data burst is timed out before it reaches the minimum required length and padding overhead must be added. Note that in this chapter, we mainly focus on case (c) representing instances when the incoming IP packet arrival rate of sub-bursts is low. Consequently, in such cases, the burst assembly approach will be timer-based, and bursts will be released prior to reaching their minimum length requirement. The padding overhead will increase the network load and can lead to increased blocking in the network.

One approach to minimize the amount of padding overhead, as well as the average end-to-end IP packet delay due to low IP packet arrival rate is to *groom* bursts. Burst grooming is defined as aggregating multiple sub-bursts with different characteristics (i.e. edge node destinations) together at the edge node and transmitting them as a single burst. In situations where the overall load is high, if there are still several sub-bursts with low arrival rate, the padding overhead for these sub-bursts can still have a significant impact on the network performance, particularly on bottleneck links. Thus, even under higher overall network loads, burst grooming may potentially improve network performance.

The problem of aggregating and routing sub-bursts together, as well as determining

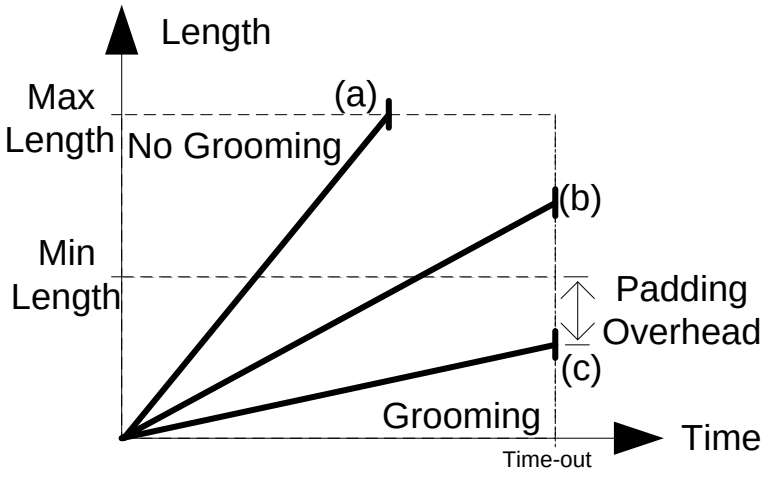


Figure 6.1. Illustrating the timer-based and threshold-based burst assembly approaches.

their wavelength assignment, is referred to as the *data burst grooming problem*. Heuristic algorithms that attempt to solve the data burst grooming problem are referred to as *burst grooming algorithms*. These algorithms differ depending on their aggregation and routing criteria. For example, issues such as which sub-bursts and how many sub-bursts can be groomed together, or how long the accumulated length of the groomed burst should be, can have significant impact on the efficiency of the grooming algorithm under different network loading conditions. We note that the general burst grooming concept can be implemented in conjunction with any given scheduling and routing algorithms.

The concept of traffic grooming has been extensively studied for various circuit-switched WDM network topologies (ring, mesh, etc.) under different traffic scenarios (static or dynamic) [139], [140], [141], [142], [143]. The basic idea in all these problems is to share *lightpaths*, defined as wavelength channels dedicated to established connections. Hence, we refer to these problems as lightpath-based grooming problem. The objective of data burst grooming in OBS over WDM networks, however, is to aggregate multiple sub-bursts to share the data burst created to satisfy a request. Data burst grooming in OBS has not received much attention in the literature. In [89] the authors consider data burst grooming

at core nodes where several sub-bursts sharing a common path can be aggregated together in order to reduce switching overhead. The aggregated sub-bursts can be separated at a downstream node prior to reaching their final destinations.

In this chapter we address the problem of data burst grooming in OBS networks. In our study, we concentrate on grooming data bursts at the edge nodes. This study is motivated by the following network constraints: (a) the data traffic through the network is bursty in nature and connections are short lived; (b) at low IP packet arrival rate instances, the core node switching time is much larger than the average IP packet size; (c) incoming IP packets passing through the network have a maximum end-to-end delay tolerance. We emphasize that under such conditions, traffic-aware assembly and routing schemes may not be efficient due to the bursty nature of the traffic. Similarly, lightpath-based grooming with static connections will not be suitable because it does not support on-demand network reconfigurability. On the other hand, dynamically reconfigurable lightpath-based grooming may not efficiently utilize the available bandwidth because data connections have short duration relative to the setup time of the lightpaths. Note that without the delay tolerance constraint, packets can stay in the assembly unit indefinitely and there will be no need for burst grooming.

The main contribution of this chapter is an edge node architecture for enabling burst grooming, as well as several data burst grooming heuristic algorithms. Using simulation we examine the performance of our proposed grooming algorithms under specific network conditions. We compare our results with those obtained without burst grooming in terms of blocking probability and average end-to-end IP packet delay. We show that our proposed burst grooming techniques lead to performance improvement when the IP traffic arrival rate is low.

The remainder of this chapter is organized as follows. In Section 6.2, we describe the proposed edge node architecture in OBS networks capable of supporting data burst grooming. Section 6.3 formulates the data burst grooming problem and provides descriptions of two proposed grooming algorithms. The performance results for each algorithm are

presented in Section 6.4. Finally, Section 6.5 concludes this chapter.

6.2 Node Architecture

The general core node architecture is described in details in [100] and [68]. We assume that the switching time for core nodes is given as τ, and that the minimum required data burst duration is defined as a function of τ: $L^{MIN} = f(\tau)$. Throughout this section, we refer to sub-bursts as the aggregated IP packets with the same edge node destination, whose total length is less than L^{MIN}. Hence, a transmitted burst can contain multiple sub-bursts.

Fig. 6.2 shows the basic architecture of an edge node supporting data burst grooming. An *ingress* edge node, which generates and transmits data bursts to core nodes, performs the following operations: (a) burst assembly: aggregating incoming IP packets with the same edge node destination (or other similar characteristics) in a virtual queue (VQ); (b) sub-burst grooming: combining multiple sub-bursts from different VQs into a single burst; (c) burst scheduling: attaching padding and preamble (framing) overhead to the bursts and scheduling them for transmission on an appropriate channel; (d) BHP generation: constructing the header packets and transmitting them prior to their corresponding data bursts.

In the *egress* path, as shown in Fig. 6.2, an egress edge node performs two basic functions: burst disassembly and IP routing. Upon receiving a data burst, the edge node initially disassembles the burst. The extracted sub-bursts, which need to be retransmitted to the downstream nodes are sent to the assembly unit, while the remaining sub-burst will be directed to the IP-routing unit. The IP-routing unit is a line card responsible for disassembling each sub-burst and sending its embedded packets to appropriate IP routers in the access layer of the network.

We assume that the total IP packet delay in the network must be less that the maximum tolerable end-to-end packet delay, denoted by T_e. Note that, in this architecture, when an incoming disassembled sub-burst requires immediate retransmission and is routed to the assembly unit, it will be treated as a timed-out sub-burst and hence, must be released immediately.

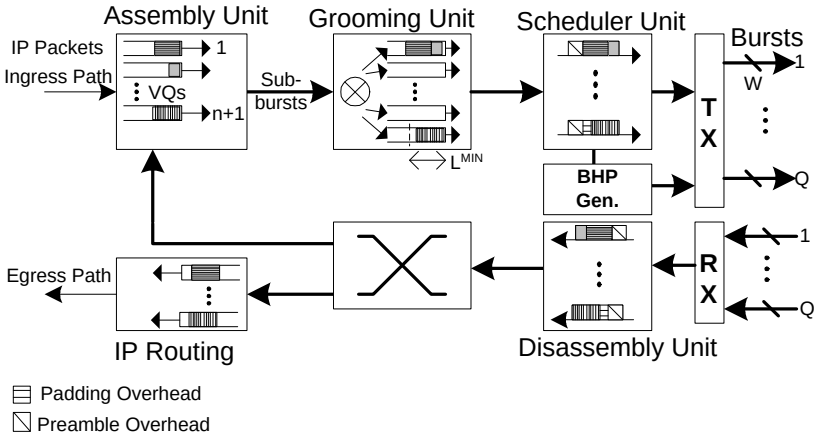


Figure 6.2. An edge node architecture supporting burst grooming with Q ports and W data channels and one control channel on each port.

6.3 Burst grooming

In this section we first introduce some basic definitions and formulate the edge node grooming problem in OBS network, and then describe our proposed grooming algorithms. A summary of notations used following sections is provided in Table 6.1.

6.3.1 Data burst grooming

We denote a sub-burst i as b_i. Each sub-burst b_i consists of multiple IP packets with the same edge node destination and can be characterized by its edge node source, destination, and length: S_{b_i}, D_{b_i}, and L_{b_i}. As soon as an IP packet with destination D_{b_i} arrives to a node, a timer is set for sub-burst b_i. The sub-burst will be released when it is timed out. The time-out value is based on the maximum end-to-end delay tolerance that IP packets can tolerate within the OBS network, denoted by T_e.[1] Therefore, the time-out value for data bursts in each virtual queue is bounded by the difference between T_e and the sum of OBS source-destination propagation delay and node processing delays, which includes the burst disassembly time at the destination node. In addition to the aforementioned parameters, each sub-burst, b_i, has a *remaining slack time*, denoted as δ_{b_i}. The remaining slack time

[1] In general, depending on their class of service, IP packets can have different T_e values. In this work we assume all IP packets belong to the same class of service and hence, the same T_e value can be applied to all IP packets.

is defined as the remaining tolerable end-to-end delay the sub-burst can tolerate before it reaches its destination.

Table 6.1. Summary of parameter definition.

Parameter	Description
$\mathbf{G} = \{b_0, b_1, ...\}$	Groomed data burst set, b_0 is the timed-out sub-burst
$\|\mathbf{G}\|$	Combined length of sub-bursts groomed together
L_G	The number of sub-bursts groomed in a single burst
b_i, L_{b_i}	Sub-burst b_i with length L_{b_i}
G^{MAX}	Max. number of sub-bursts allowed to be groomed
T_e	Max. tolerable end-to-end delay for IP packet class
L^{MIN}	Minimum required burst size
δ_{b_0}	Stack time of sub-burst b_0
$\Delta(b_i, b_0)$	Route deflection distance
$\Psi(b_i, \mathbf{G})$	Relative routing and padding overhead
$\Upsilon(b_i, b_0)$	Relative routing overhead of b_i
S_{b_i}, D_{b_i}	Edge node source and destination of burst b_i
$H_p(S_{b_i}, D_{b_i})$	Shortest phy. distance between (S_{b_i}, D_{b_i})
ρ_{net}, ρ_{act}	Average IP and data burst traffic load, respectively

We represent a groomed data burst by $\mathbf{G} = \{b_0, b_1, b_2, ...\}$, which is constructed by aggregating a number of sub-bursts with different destinations. We consider the first element (sub-burst) in the grooming set (b_0) as the timed-out sub-burst, which must be routed on a single hop shortest-path. Hence, the first hop for all sub-bursts in **G** will be the node corresponding to the destination D_{b_0}. In our notation $|\mathbf{G}|$ and L_G indicate the number of sub-bursts groomed together and their combined length, respectively. Clearly if $|\mathbf{G}|$=1 no grooming has been performed and $L_G = L_{b_0}$. Furthermore, we refer to G^{MAX} as the maximum number of sub-bursts which are allowed to be groomed together prior to transmission, hence $|\mathbf{G}| \leq G^{MAX}$.

We define the *hop-delay* as the delay time imposed on an incoming sub-burst due to electronic processing. In our study, we only consider the maximum hop-delay, expressed as T_h, and assume it is the same for all nodes. It is clear that the timed out sub-burst can only be groomed with any other sub-burst, b_i, whose remaining slack time satisfies the following expression:

$$T_p(S_{b_0}, D_{b_0}) + T_p(D_{b_0}, D_{b_i}) + T_h \leq \delta_{b_i} \leq T_e. \tag{6.1}$$

In the above expression, $T_p(s, d)$ is the propagation delay from node s to node d. Note that δ_b for any given sub-burst is bounded by T_e.

When **G** reaches its first destination node, D_{b_0}, sub-burst b_0 is dropped. Then, each remaining sub-burst, b_i, in the grooming set **G**, is directed to its proper virtual queue and its slack time is reduced by $T_h + T_p(S_{b_0}, D_{b_0})$. Incoming sub-bursts may be aggregated with the existing IP packets waiting in the corresponding virtual queue. In this case, the remaining slack time of the *combined* sub-burst is set to the remaining slack time of the earliest packet in the queue.

We illustrate the above concepts using the example shown in Fig. 6.3. In this example, The sub-burst at Node 1 going to Node 3 is timed out and it is groomed with another sub-burst with destination Node 7, in order to meet the minimum length requirement. At Node 3, the sub-burst with destination Node 3 is dropped. The remaining sub-burst going to Node 7 will be groomed with another sub-burst with destination Node 6. At Node 7, the sub-burst going to Node 6 is sent to the proper virtual queue and combined will all existing IP packets in the queue. When the timer is expired, the combined sub-burst going to Node 6 must be transmitted. In this case, since the minimum length is not met, padding overhead is added.

When a sub-burst b_0 is timed out, the burst grooming algorithm finds the appropriate **G** ($b_0 \in$ **G**) among all possible grooming combinations. Selection of the grooming set is based on the optimization objective of the grooming algorithm. Aggregating multiple sub-bursts reduces the *padding overhead*, which in turn, can improve the blocking probability. However, this can potentially result in routing the groomed sub-bursts over longer physical paths. This phenomena, referred as the *routing overhead*, can impact the network throughput.

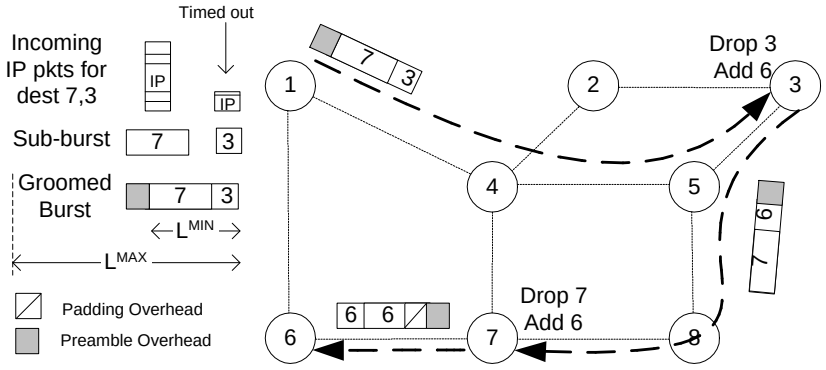


Figure 6.3. A simple network carrying groomed data bursts.

For example, consider Fig. 6.3, where at Node 1 the timed-out sub-burst going to Node 3 is groomed with the sub-burst going to Node 7. We denote the shortest physical hop distance between node pair (s, d) by $H_p(s, d)$. In this case, the sub-burst going to Node 7, will be traveling over $H_p(1, 3) + H_p(3, 7) = 3 + 3 = 6$ physical hops, whereas the shortest physical hop distance between Node 1 and Node 7 is 2: $H_p(1, 7) = 2$. This example demonstrates that simple greedy aggregation of sub-bursts can have adverse effects. Consequently, an effective grooming policy must minimize both the padding and the routing overhead while minimizing additional hop-delay.

It is evident in the above example that a potential drawback of burst grooming is the increase in number of electrical-to-optical converter/transmitter as incoming groomed sub-bursts must be re-transmitted from intermediate nodes to their final edge node destinations. Furthermore, burst grooming can result in higher buffering requirements at intermediate nodes. We defer these issues until later studies.

6.3.2 Problem formulation

In an OBS mesh network, data burst grooming can be performed at the edge node. In this case, each individual edge node must decide how to aggregate individual sub-bursts with durations smaller than the minimum length requirement, in order to optimize the throughput and reduce the probability of burst dropping. Hence, we can formulate the data burst

grooming problem at the edge node as follows. *Given* the entire network information (including the physical network topology and full routing knowledge between all node pairs), the minimum required data burst duration, the maximum end-to-end delay that each class of IP packet can tolerate, and a timed-out sub-burst with a length smaller than the minimum required length has timed-out, ***find*** any available sub-burst, b_i, which can be aggregated with the timed-out sub-burst, b_0, in order to minimize blocking probability.

We consider the following assumptions: all edge nodes have full grooming capability and equipped with full wavelength converters; all incoming IP packets have arbitrary lengths and a single destination; data bursts with durations shorter than the minimum burst length requirement will be subject to padding overhead; all IP packets in a virtual queue must be transmitted together. In addition, in this study, we focus on networks with low IP traffic arrival rate; thus, only a timer-based triggering scheme is assumed. We assume source routing, where the source edge node knows the entire path for all sub-bursts.

6.3.3 Description of grooming algorithms

An intuitive approach to reduce IP packet blocking probability is to develop effective grooming algorithms in order to reduce overall network overhead. The efficiency of grooming algorithm can be affected by several parameters, including the number of sub-bursts, which can be groomed together, the accumulated length of the groomed sub-bursts, and the way groomed sub-bursts with different destinations are routed. These parameters can have conflicting impacts under different network conditions. For example, at moderate loads, having fewer constraints on the above parameters may considerably reduce the network overhead, resulting in higher network throughput. On the contrary, at higher loads, asserting no constraints on the above parameters may notably alter the traffic characteristics and increase traffic burstiness, resulting in higher packet blocking.

We distinguish grooming algorithms by the way the source node calculates the padding and routing overheads due to burst grooming. Since the source node has no knowledge about the traffic between other node pairs, its padding overhead calculations are based on

worst case *local* estimations. In our study, we consider two grooming algorithms: No-routing-overhead *(NoRO)* and Minimum-total-padding-overhead *(MinTO)*.

No-routing-overhead algorithm (NoRO): The main objective in this algorithm is to ensure no routing overhead is added as sub-bursts are groomed together. In practice, this leads to routing all sub-bursts through their shortest-cost path. Depending on the cost metric, the shortest-cost path can be, for example, based on physical hop distance or total link distance. Without, loss of generality, in this section we consider the physical hop distance between node pair as the cost metric and refer to it as the *shortest path*. Note that NoRO does not distinguish between alternative shortest-cost paths or interdependencies between them, as long as the cost remains the same.

The routing overhead for a sub-burst b_i when groomed with b_0, can be quantified using the relative routing overhead, $\Upsilon(b_i, b_0)$, which is defined as

$$\Upsilon(b_i, b_0) = \frac{H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_i})}. \tag{6.2}$$

Sub-burst b_i and b_0 can only be groomed if $\Upsilon(b_i, b_0) = 1$, indicating that the destination of the timed-out sub-burst, D_{b_0}, is on the shortest path to the destination of the groomed sub-burst, D_{b_i}.

The details of the NoRO grooming algorithm as sub-burst b_0 with length L_{b_0} is timed out are shown in Fig. 6.4. We denote all available sub-bursts (excluding b_0) in virtual queues as set $\mathbf{S} = \{b_1, ..., b_i, ...\}$. Note that the NoRO algorithm continues to groom b_0 with other sub-bursts until the combined length of the groomed burst, L_G, is larger than L^{MIN} or the number of groomed sub-bursts, $|\mathbf{G}|$, has exceeded G^{MAX}.

Minimum-total-overhead algorithm (MinTO): The main objective of this algorithm is to reduce the combined routing and padding overheads by grooming multiple sub-bursts together. In practice, this leads to relaxing the no-routing-overhead constraint and allowing sub-bursts to travel through additional physical hops, when compared to their shortest path, before reaching their edge node destinations. The combined routing and padding overheads

Initialization: $\mathbf{G} = \{b_0\}$, $\mathbf{S} = \{b_1, ..., b_i, ...\}$, while $L_G < L^{MIN}$, $|\mathbf{G}| < G^{MAX}$ and $\mathbf{S} \neq \emptyset$
- Select $b_i \in \mathbf{S}$ with the largest length such that δ_{b_i} satisfies Eqn. (6.1) and $\Upsilon(b_i, b_0) = 1$
- if b_i exists, move b_i from **S** to **G** update L_G and $|\mathbf{G}|$
- else $\mathbf{S} = \emptyset$

end while

Figure 6.4. No-routing-overhead algorithm (NoRO).

for a sub-burst b_i, if groomed with set a set of sub-bursts **G**, can be quantified by the *relative routing and padding overhead*, $\Psi(b_i, \mathbf{G})$, which is defined as

$$\Psi(b_i, \mathbf{G}) = \frac{\hbar(L_G + L_{b_i}) \cdot H_p(S_{b_0}, D_{b_0}) + \sum_{b_j \in \mathbf{G} \cup b_i}^{b_j \neq b_0} \hbar(L_{b_j}) \cdot H_p(D_{b_0}, D_{b_j})}{\sum_{b_j \in \mathbf{G} \cup b_i} \hbar(L_{b_j}) \cdot H_p(S_{b_0}, D_{b_j}).}, \tag{6.3}$$

In the above expression $\hbar(x) = max(L^{MIN}, x)$ and $\mathbf{G} \cup b_i = \{b_0, b_i\}$ if $\mathbf{G} = \{b_0\}$. Note that the necessary condition for b_i to be groomed with set **G**, where $b_0 \in \mathbf{G}$, is $\Psi(b_i, \mathbf{G}) < 1$.

The additional physical hops a sub-burst b_i, when groomed with $b_0 \in \mathbf{G}$, must traverse before it reaches its edge node destination, is referred to as *route deflection distance* and we define it by

$$\Delta(b_i, b_0) = (H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})) - H_p(S_{b_0}, D_{b_i}). \tag{6.4}$$

For example, referring to Fig. 6.3, the sub-burst going to Node 7 from Node 1 will have to tolerate a route deflection distance of $\Delta = 6 - 2 = 4$.

Clearly, if Δ is limited to zero no route deflection will be allowed and all sub-bursts must traverse along their shortest paths. Note that having $\Delta = 0$ also implies no routing overhead: $\Upsilon = 1$. Details of the MinTO grooming algorithm as sub-burst b_0 with length L_{b_0} is timed out are shown in Fig. 6.5.

.

6.3.4 Algorithm analysis

In this section we take a closer look at the MinTO algorithm and examine its performance under three different loading conditions. For simplicity we assume that maximum number

Initialization: **G** $= \{b_0\}$, **S** $= \{b_1, ..., b_i, ...\}$ while $L_G < L^{MIN}$, $|\mathbf{G}| < G^{MAX}$ and **S** $\neq \emptyset$
- Select $b_i \in$**S** with smallest $\Psi(b_i, \mathbf{G}) < 1$ and largest length such that δ_{b_i} satisfies Eqn. (6.1) and $\Delta(b_i, b_0)$ <max. allowable route deflection
- if b_i exists, move b_i from **S** to **G** update L_G and $|\mathbf{G}|$
- else **S** $= \emptyset$

end while

Figure 6.5. Minimum-total-overhead algorithm (MinTO).

of sub-bursts that can be groomed in a single burst is two, $G^{MAX} = 2$.

(a) *Light loads* ($L_G, L_{b_0}, L_{b_i} < L^{MIN}$): In this case (6.3) will be reduced to

$$\Psi(b_i, \mathbf{G}) = \frac{H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) + H_p(S_{b_0}, D_{b_i})}, \tag{6.5}$$

Using (6.4), the necessary condition for $\Psi(b_i, \mathbf{G}) < 1$, indicating b_i can be groomed with $b_0 \in$ **G**, is

$$\Delta(b_0, b_i) \leq H_p(S_{b_0}, D_{b_0}). \tag{6.6}$$

If the route deflection distance is zero, $\Delta = 0$, under the low loading assumption, (6.5) is reduced to

$$\Psi(b_i, \mathbf{G}) = \frac{H_p(S_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) + H_p(S_{b_0}, D_{b_i})} < 1. \tag{6.7}$$

In this case, $\Psi(b_i, \mathbf{G})$ will be smaller in value for sub-bursts b_i with shorter hop distance from S_{b_0} to D_{b_i}: $H_p(S_{b_0}, D_{b_i})$.

(b) *Moderate loads* ($L_G \geq L^{MIN}, L_{b_0}, L_{b_i} < L^{MIN}$): In this case (6.3) will be reduced to

$$\Psi(b_i, \mathbf{G}) = \frac{H_p(S_{b_0}, D_{b_0}) \cdot (L_G/L^{MIN}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) + H_p(S_{b_0}, D_{b_i})}. \tag{6.8}$$

Rewriting the above expression in terms of Δ, we obtain

$$\Delta(b_0, b_i) \leq H_p(S_{b_0}, D_{b_0})(L_G/L^{MIN}). \tag{6.9}$$

Comparing (6.5) and (6.8), suggests that as long as $L_G < L^{MIN}$ and $H_p(D_{b_0}, D_{b_i}) < H_p(S_{b_0}, D_{b_i})$, the timed-out sub-burst can be groomed with b_i. However, as the load increases and $L_G > L^{MIN}$, fewer burst grooming can be expected.

(c) *Higher loads* ($L_{b_i} \approx L_G \geq L^{MIN}, L_{b_0} < L^{MIN}$): In this case (6.3) can be expressed as

$$\Psi(b_i, \mathbf{G}) = \frac{H_p(S_{b_0}, D_{b_0}) + H_p(D_{b_0}, D_{b_i})}{H_p(S_{b_0}, D_{b_0}) \cdot (L^{MIN}/L^G) + H_p(S_{b_0}, D_{b_i})}. \tag{6.10}$$

Using the definition for Δ, the above expression can be rewritten as

$$\Delta(b_0, b_i) \leq H_p(S_{b_0}, D_{b_0})(L^{MIN}/L_G), \tag{6.11}$$

where $L^{MIN}/L_G \leq 1$.

In the above discussion we can clearly see that, in order to minimize routing and padding overhead, MinTO continuously attempts to groom multiple small sub-bursts, whose destinations are closest to D_{b_0}. On the contrary, the NoRO algorithm mainly attempts to find the largest available sub-burst traveling along the timed-out sub-burst's path. An interesting observation in comparing (6.6), (6.9), and (6.11) is that as the network load increases, smaller route deflection distance will be allowed and hence, less grooming opportunities will be provided by MinTO. Furthermore, the above relationships show that under certain network conditions, MinTO reduces the overall overhead in the network by introducing minimum routing overhead, $\Delta \neq 0$. This is different from NoRO, which aggressively attempts to search for the largest available sub-bursts to be groomed, regardless of the network load.

We illustrate the behavior of the NoRO and MinTO using the example shown in Fig. 6.6, where a 5-node network with a single optical channel between each node pair is considered. We assume at Node a sub-burst b_y is timed out and can be groomed with one of the available sub-bursts: b_w, b_x, or b_z. Using the NoRO algorithm, if we groom sub-burst b_y with b_z, the lowest Υ value can be obtained. On the other hand, using the MinTO algorithm, the grooming choice changes depending on the length ratio of the available sub-bursts, namely, b_w, b_x, and b_z, over L^{MIN}. For example, assuming the length of b_z is much

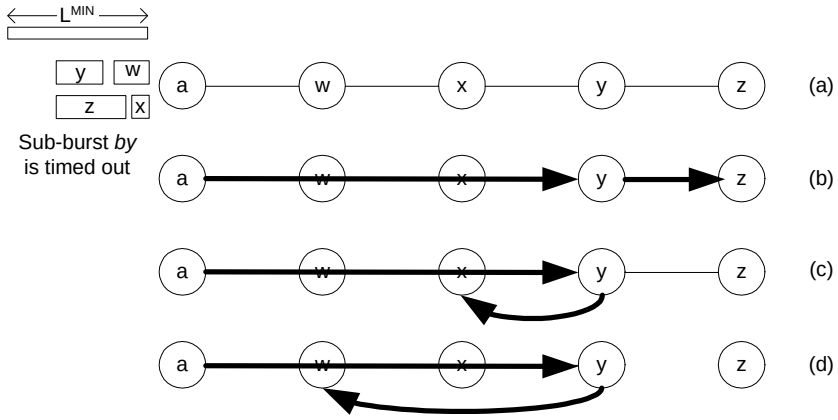


Figure 6.6. An example of a 5-node network where sub-burst b_y going to Node y is timed out and it can be groomed with any one of the available sub-bursts: b_w, b_x, or b_z. Note that we assume the size of the grooming set is limited to $G^{MAX} = 2$.

larger than L_{b_x} and L_{b_w}, the value of Ψ for b_x, b_z and b_w varies depending on the length of the timed-out sub-burst, b_y, as shown in Fig. 6.7. It can be seen, that for high values of L_{b_y}/L^{MIN}, $\Psi(b_x, b_y)$ will be the smallest and hence, b_x will be selected to be groomed with b_y. This shows, that under special circumstances, the MinTO algorithm prefers to groom with an available sub-burst which results in larger route deflection distance.

Fig. 6.8 demonstrates the range where the value of $\Psi(b_w, b_y)$, with $\Delta(b_0, b_w) = 2$ is smaller than $\Psi(b_x, b_y)$ with $\Delta(b_0, b_x) = 1$.

6.4 Performance results

In this section we present the simulation results obtained by implementing the NoRO and MinTO algorithms. We have chosen the NSFnet backbone, shown in Fig. 6.9, as our test network. In this network, we assume each link is bi-directional with a fiber in each direction. Our simulation model was developed based on the following assumptions: IP packet arrivals into the OBS network are Poisson with λ denoting their arrival rate and they are uniformly distributed over all sender-receiver pairs; IP packet length is fixed with 1250 bytes; the maximum end-to-end IP packet delay tolerance is 50 ms; the switching time at the core node is 250 μs, requiring a minimum burst length of $L^{MIN} = 250$ packets; each

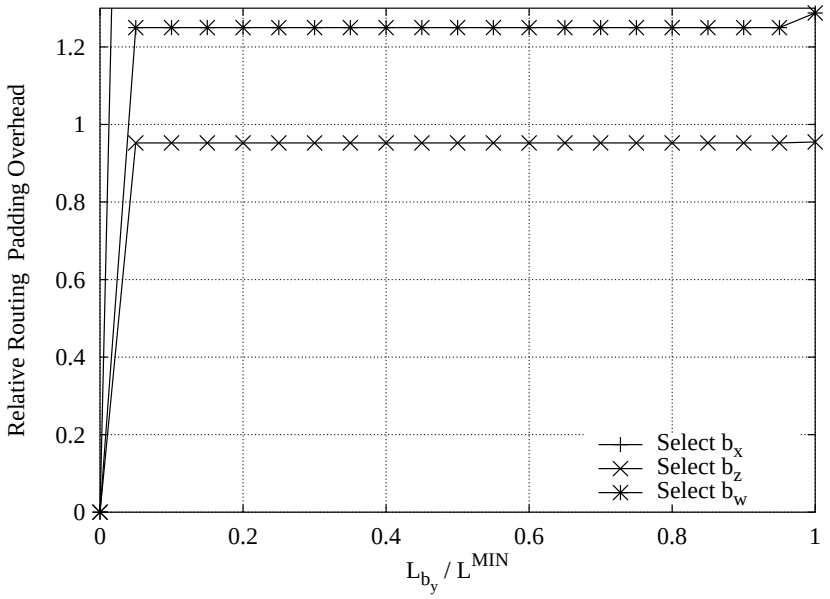


Figure 6.7. Calculating the minimum routing and padding overhead for $\mathbf{G} = \{b_y, b_x\}$, $\{b_y, b_z\}$, and $\{b_y, b_w\}$ as a function of L_{b_y}.

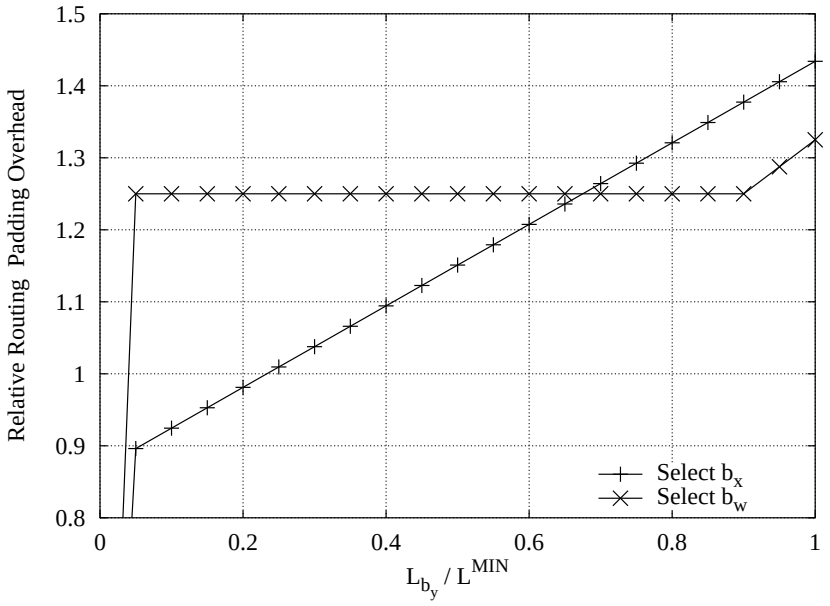


Figure 6.8. Calculating minimum routing and padding overhead for $\mathbf{G} = \{b_y, b_x\}$ and $\{b_y, b_w\}$ as a function of L_{b_y}.

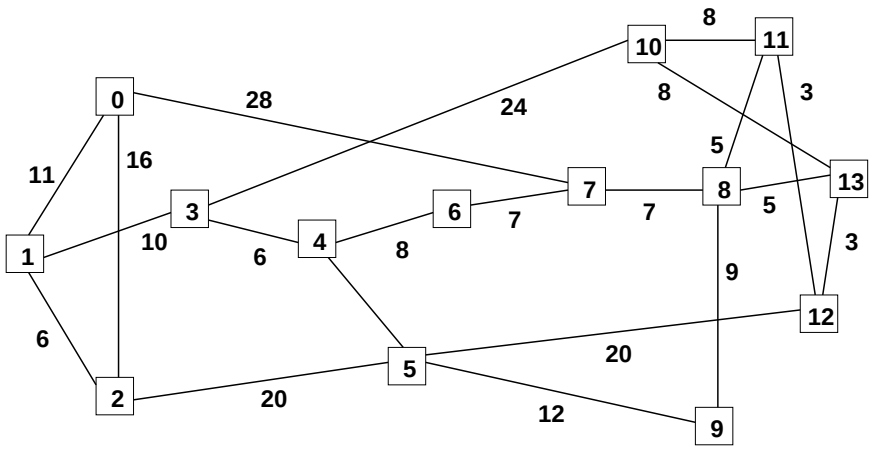


Figure 6.9. The NSF network with 14 nodes.

data burst can carry maximum of 2500 IP packets; each switch port has 8 wavelengths, each of which has a transmission rate of 10 Gbps. We also assume all nodes support data burst grooming capacity and are equipped with full wavelength converters. We adopt the latest available unscheduled channel (LAUC) algorithm to schedule data bursts at the core nodes. Furthermore, we only consider timer-based assembly and assume all sub-bursts can be groomed as long as their accumulated length is less than the minimum required length.

In our simulation study, we focus on traffic load scenarios where sub-bursts typically time out before they reach their minimum required length and hence, the mean burst length is less than L^{MIN}. Throughout this section we refer to the offered IP packet load into the network as *load*, denoted by ρ. In our results, we focus on two basic performance metrics: IP packet blocking probability and average end-to-end IP packet delay. We define the former as the ratio of the number of IP packets which did *not* reach their destination over the total number of incoming IP packets.

In our C-based simulation model we used confidence interval accuracy as the controlling factor. For each case of interest, the simulation was run until a confidence interval level of 90% was observed and an acceptably tight confidence interval (5%) were achieved. Calculations of the confidence interval were based on the variance within the collected observations [108].

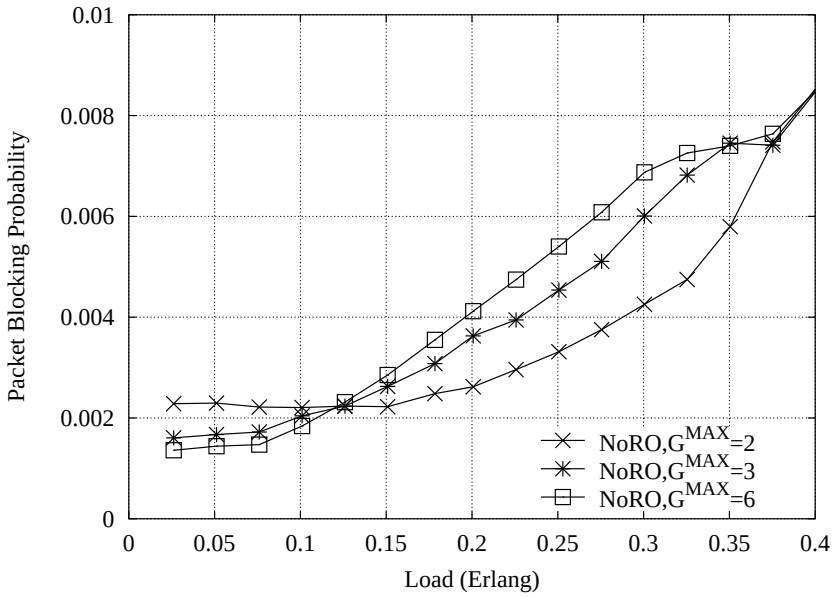


Figure 6.10. Packet blocking probability using NoRO for $G^{MAX} = 2$, 3, and 6.

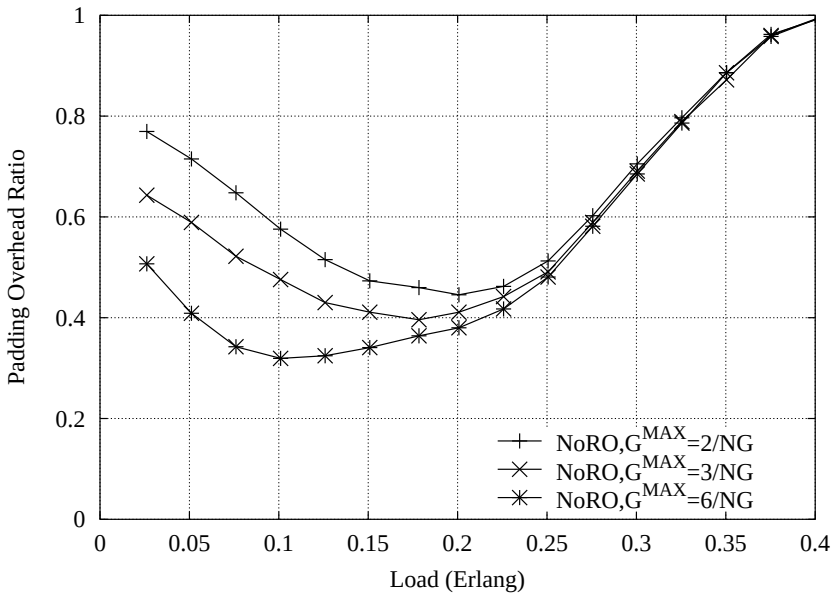


Figure 6.11. Padding overhead ratio over the OBS network using NoRO for $G^{MAX} = 2$, 3 and 6

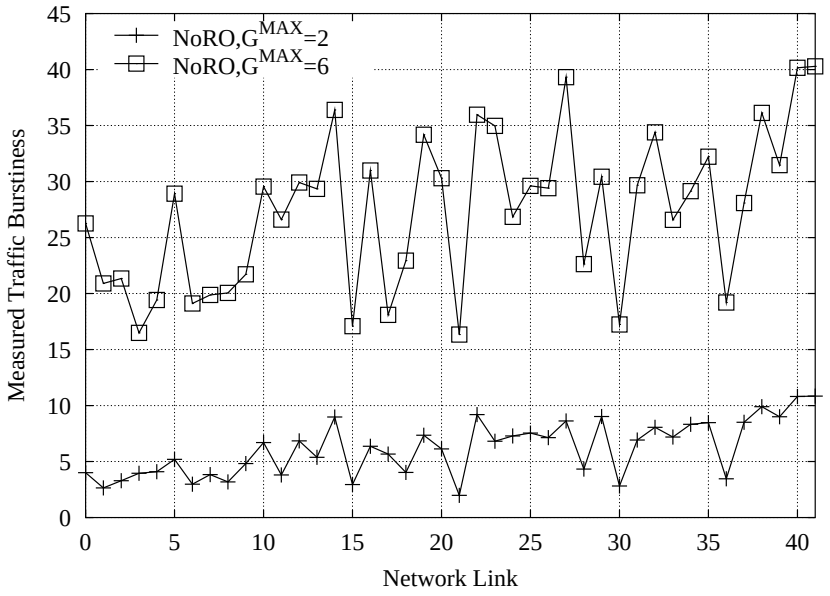


Figure 6.12. Traffic burstiness measured on each switch egress port at $\rho = 0.25$ for $G^{MAX} = 2$ and 6 using NoRO.

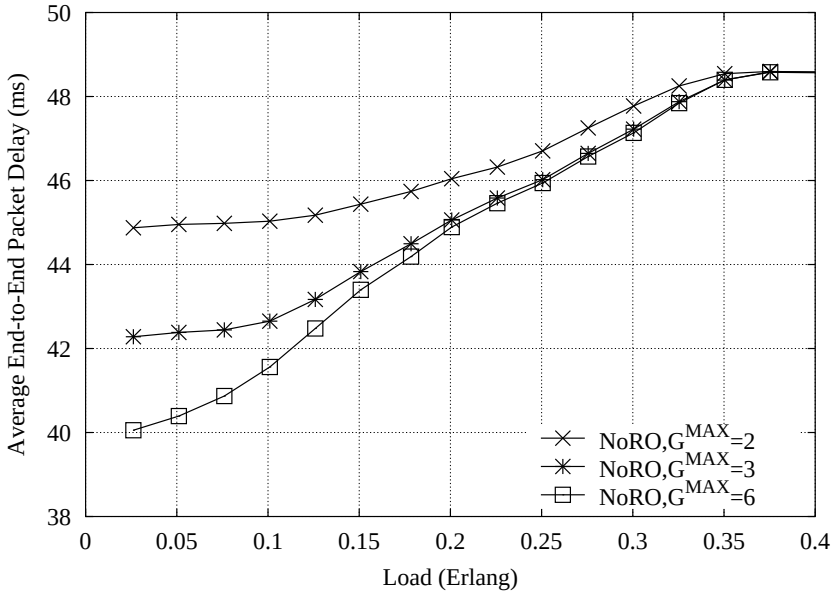


Figure 6.13. Average end-to-end packet delay (ms) using NoRO for $G^{MAX} = 2$, 3, and 6.

6.4.1 Characterizing the NoRO algorithm

Fig. 6.10 shows the performance of the NoRO grooming algorithm for $G^{MAX} = 2$, 3, and 6. As this figure suggests, under *light* loads ($\rho < 0.1$), allowing more sub-bursts to be groomed together results in lower packet blocking probability. Note that in our simulation, further increase in $G^{MAX} > 6$, does not result in further performance improvement. This is because there is no more sub-burst available in the virtual queues.

At *moderate* loads ($0.1 \leq \rho \leq 0.36$), the IP packet blocking probability increases for higher G^{MAX} values. As the load continues to increase, only a small percentage of sub-bursts will be shorter than L^{MIN} and hence, less grooming will take place. Eventually, at *higher* loads ($\rho > 0.36$), no grooming will be performed and, as Fig. 6.10 suggests, the blocking probability for different G^{MAX} values will become the same.

A surprising observation in Fig. 6.10 is that at moderate loads, as more sub-bursts are allowed to be groomed, the packet blocking probability increases. To understand this behavior, we examine two basic traffic characteristics, namely the padding overhead ratio and traffic burstiness. The former is defined as the ratio of total padding overhead with and without grooming. Clearly, having smaller padding overhead ratio implies higher link utilization compared to the case with no grooming.

We define traffic burstiness over each switch egress port i (or unidirectional link between a node pair) as variation of burst load in time interval s [144] and express it as

$$\beta_i(s) = \sqrt{\sigma_i^2(s)/\mu_i^2(s)}, \tag{6.12}$$

where $\sigma_i^2(s)$ and $\mu_i(s)$ are the variance and mean of the burst load measured on link i over some time period s, respectively. Burst load is defined as the product of burst arrival rate and mean burst length. Assuming the entire simulation period is $T_{sim} = m \cdot s$, with m discrete intervals of s, we will have $\mu_i(T_{sim}) = \sum_{k=0}^{m-1} v_i^k(s)/m = E\{v_i(s)\}$, where $v_i(s)$ is the burst load measured on link i over a time interval s. Similarly, $\sigma_i(T_{sim}) = \sqrt{E\{v_i(s)^2\} - E^2\{v_i(s)\}}$.

Fig. 6.11 indicates that at light loads, having larger G^{MAX} value can considerably reduce the padding overhead ratio. However, as the load increases, the padding overhead ratios for different values of G^{MAX} tend to become the same and approach one. Note that in Fig. 6.11 indicates that, at moderate loads the grooming algorithm results in minimum padding overhead ratio. This is attributed to the fact that at light loads, fewer and smaller sub-bursts are available to be groomed. On the other hand, at higher loads, fewer sub-bursts require padding and hence, grooming impact is minimized.

Fig. 6.12 compares the traffic burstiness, defined in Eqn. (6.12), on unidirectional links 1 through 42 for $G^{MAX} = 2$ and 6 when $\rho = 0.25$. This figure shows that as G^{MAX} increases from 2 to 6, the traffic burstiness increases as well. Increasing traffic burstiness results in higher peak burst load, leading to higher link congestion and blocking probability. Similar results can be obtained until $\rho \approx 0.36$, where grooming impact starts diminishing.

Based on the above traffic characteristics, we observe that at light loads the padding overhead ratio of $G^{MAX} = 6$ is significantly lower than that of $G^{MAX} = 2$. Consequently, the link utilization when $G^{MAX} = 6$ will be higher, leading to lower blocking probability. As the load increases, the difference between the padding overhead ratios for $G^{MAX} = 2$ and 6 is reduced and the traffic burstiness becomes the dominant factor. Hence, the blocking probability for $G^{MAX} = 6$ will be higher than $G^{MAX} = 2$.

The average end-to-end packet delay obtained from NoRO is shown in Fig. 6.13. As G^{MAX} increases, lower average delay can be achieved. This is due to the fact that by allowing higher number of sub-bursts to be groomed in a single burst, fewer sub-bursts will have to wait until they are timed out.

6.4.2 Characterizing the MinTO algorithm

The overall performance of MinTO in terms of packet blocking probability and average end-to-end packet delay, follows similar trends we described for NoRO, as shown in Fig. 6.14 and 6.15. A major issue with MinTO, however, is that it can potentially send some groomed sub-bursts through longer paths before reaching their destinations. Consequently,

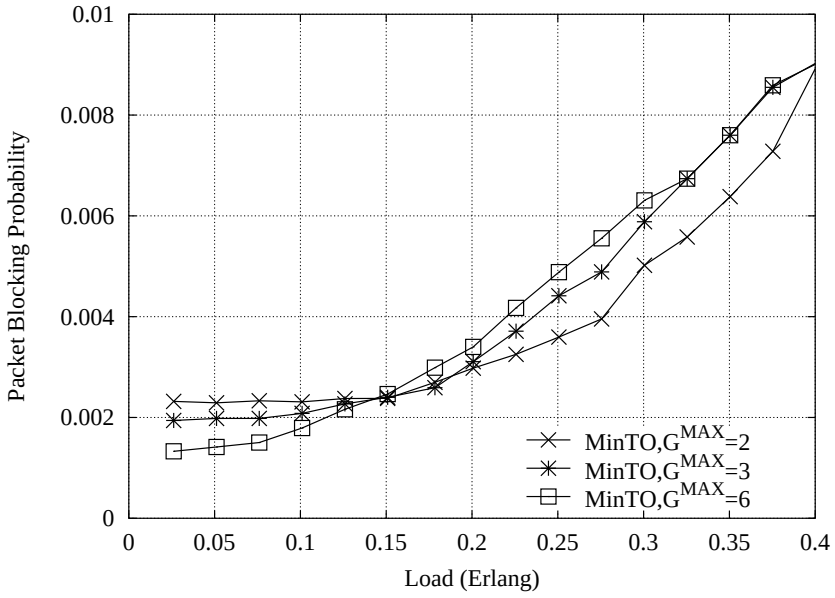


Figure 6.14. IP packet blocking probability using MinTO with different G^{MAX} values: 2, 3 and 6.

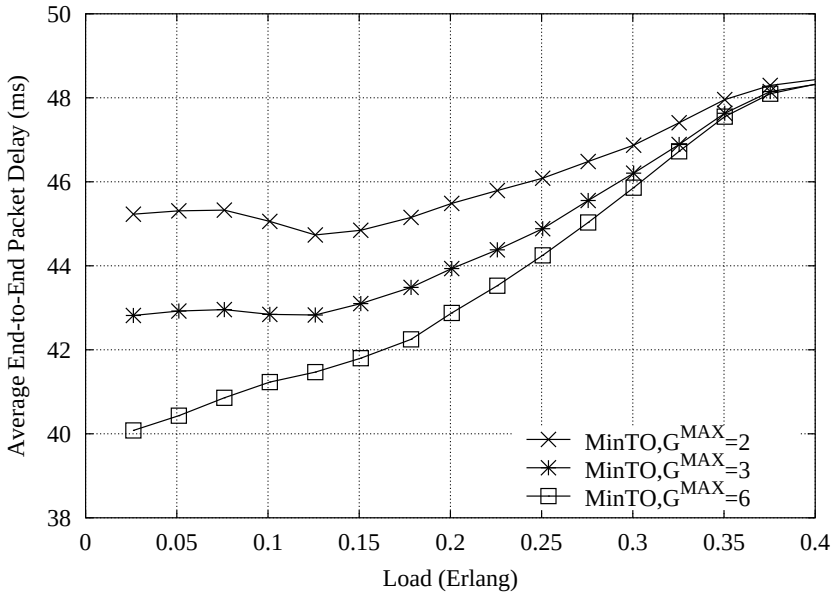


Figure 6.15. Average end-to-end IP packet delay (ms) using MinTO for $G^{MAX} = 2$, 3, and 6.

such sub-bursts will be more vulnerable to blocking. In fact, further simulation shows that at light loads, sub-bursts experience an average route deflection distance of $\Delta \approx 1.9$. As the load increases, Δ tends to become smaller, which can be verified by comparing Eqn. (6.5) and Eqn. (6.8). This effect can also be observed in Fig. 6.15. When $G^{MAX} = 2$, the average number of sub-bursts groomed together remains the same for all loads and the average end-to-end packet delay at $\rho = 0.05$ is slightly larger than when $\rho = 0.13$. However, when $G^{MAX} = 6$, this effect is not evident because the average number of sub-bursts groomed together changes.

One way to avoid excessive route deflection is to impose an upper bound on the maximum route deflection distance, for example, $\Delta \leq 1$. Our simulation results showed that under such constraint, at higher loads, slightly lower packet blocking can be achieved, which are consistent with our analysis in Sec. 6.3.4. The tradeoff for such constraint is the higher average end-to-end packet delay due to limited grooming opportunities. In the remainder of this section we consider MinTO where $\Delta \geq 0$.

6.4.3 Grooming algorithm comparison

In this section we compare the performance of NoRO and MinTO with the case when no grooming is applied. We start by examining the average number of sub-bursts groomed obtained by each algorithm as the load changes. Then, we demonstrate how the average number of sub-bursts groomed impacts the performance metrics.

Fig. 6.16 shows that when $G^{MAX} = 6$, at light loads, MinTO is less *aggressive* and provides fewer grooming opportunities compared to NoRO. This is due to the fact that in MinTO burst grooming depends on the latest value of $\Psi(b_i, \mathbf{G})$ and the sub-bursts that have already been included in $\mathbf{G}$. As the load increases, there are more sub-bursts available. Hence, MinTO provides more grooming opportunities by relaxing the routing overhead constraint and allowing route deflection distance. When $G^{MAX} = 2$, the average number of sub-bursts groomed will be similar for both algorithms. However, NoRO is more aggressive because it tends to select the largest available sub-bursts for grooming, as described in

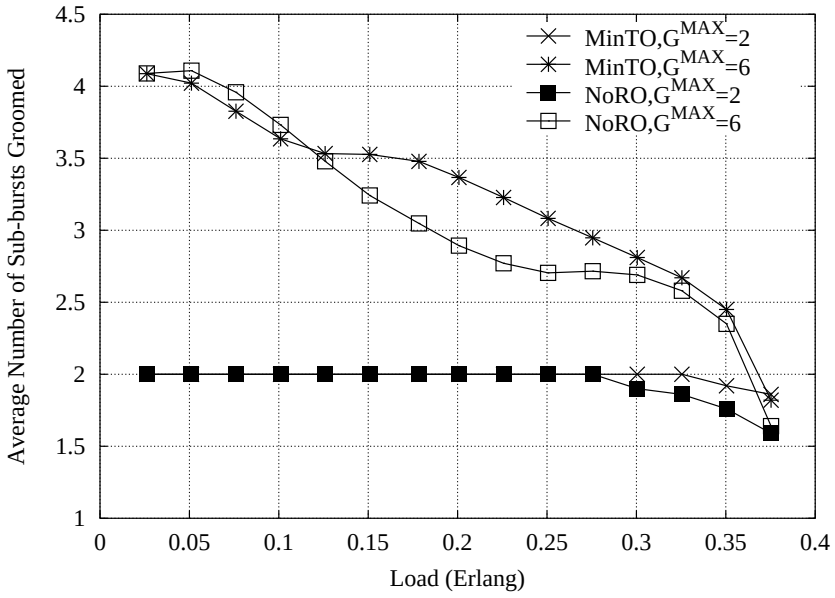


Figure 6.16. Comparing the average number of sub-bursts groomed in a single burst using NoRO and MinTO for $G^{MAX} = 2$ and 6.

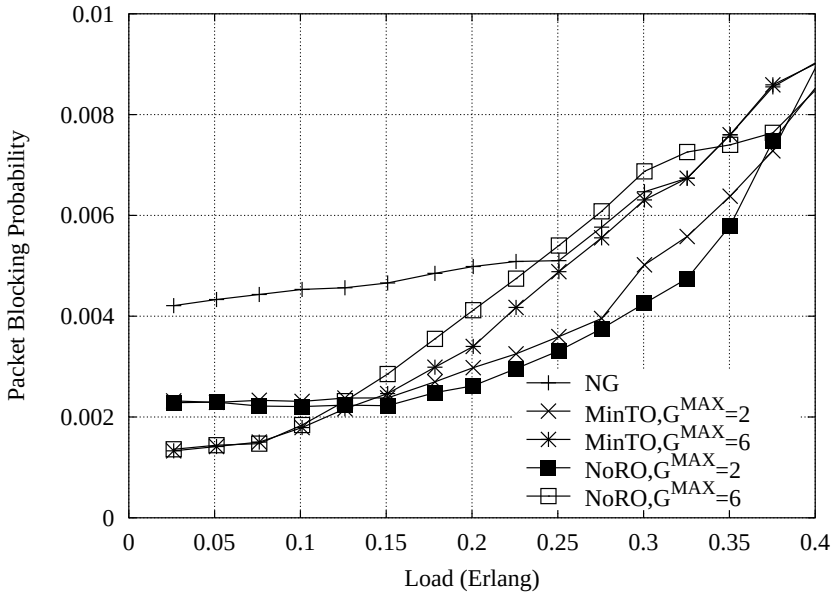


Figure 6.17. Comparing the packet blocking probability using NoRO and MinTO for $G^{MAX} = 2$ and 6.

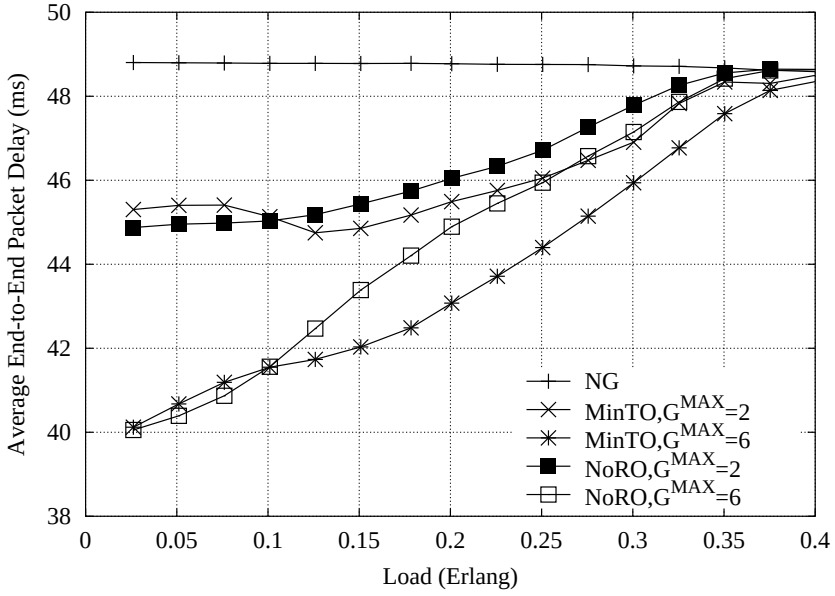


Figure 6.18. Comparing the average end-to-end packet delay (ms) using NoRO and MinTO for $G^{MAX} = 2$ and 6.

Fig. 6.4.

Fig. 6.17 shows the packet blocking probability obtained by implementing the NoRO and MinTO for $G^{MAX} = 1$, 2 and 6. At light loads, the more aggressive grooming approach, i.e. NoRO, with higher G^{MAX}, provides slightly lower blocking probability. At higher loads, however, a more aggressive grooming approach results in higher traffic burstiness on links, as shown in Fig. 6.12. Consequently, NoRO with $G^{MAX} = 2$ outperforms MinTO with $G^{MAX} = 6$ in terms of packet blocking probability. Recall that even limited grooming, $G^{MAX} = 2$, can considerably reduce the padding overhead ratio and hence, improve the overall blocking probability.

Fig. 6.18 shows that, in general, the average end-to-end packet delay due to grooming is much less than the case in which no grooming is implemented. Furthermore, the relative performance of MinTO and NoRO consistently follows the algorithm's grooming aggressiveness, as shown in Fig. 6.16. That is, more grooming results in lower end-to-end packet delay.

The aforementioned results demonstrate that MinTO and NoRO perform differently depending on the load. In general, grooming higher number of sub-bursts together can

considerably improve the average end-to-end packet delay. Burst grooming can also improve packet blocking probability throughout the network at moderate loads. However, at higher loads, depending on network constrains, such as L^{MIN} and T_e, limited grooming must be considered to reduce the padding overhead and hence, to reduce the packet blocking probability. An adaptive approach, which aggressively grooms sub-bursts at low loads and gradually decreases G^{MAX} as the load increases, can improve the overall network performance in terms of both blocking probability and average end-to-end packet delay.

6.4.4 Performance of NoRO under different network parameters

In this section we investigate the performance of the grooming algorithms as the maximum tolerable end-to-end packet delay, T_e, and the minimum burst length requirement, L^{MIN}, vary. Since both NoRO and MinTO behave similarly under such changes, we only focus on performance of the NoRO grooming algorithm.

In general, for a given switching time and load, as T_e decreases, data bursts time out earlier and hence, the average number of IP packets aggregated in each burst tends to become smaller. Consequently, more padding overhead will be generated and higher packet blocking probability is expected. Fig. 6.19 shows the packet blocking probability using NoRO with $G^{MAX} = 2$ for $T_e = 50$ and 60 ms. This figure suggests that for a given load and switching time, NoRO becomes more effective in terms of lowering the packet blocking probability as T_e is reduced. This implies that burst grooming can particularly benefit IP packets with lower end-to-end delay tolerance.

Fig. 6.20 shows the percentage performance improvement of NoRO with $G^{MAX} = 2$ compared to when no grooming is implemented, as L^{MIN} changes from 250 to 350. As L^{MIN} increases, NoRO becomes more effective in terms of lowering the blocking probability for higher loads. Similarly, our simulation results confirmed that for a given load and T_e, as L^{MIN} increases, burst grooming can become more effective in terms of lowering the average end-to-end packet delay.

We also examined the impact of burst grooming when *no* wavelength converters were

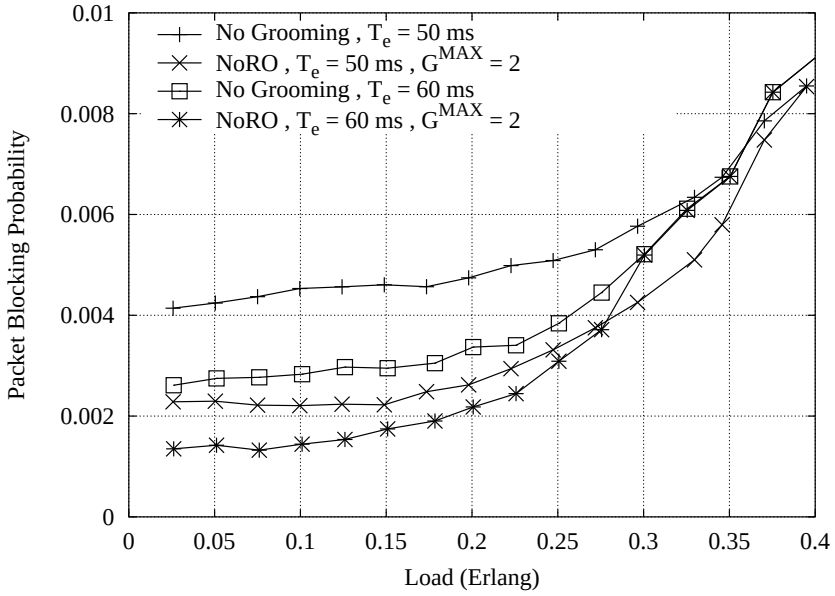


Figure 6.19. Comparing the packet blocking probability using NoRO and no grooming for $G^{MAX} = 2$, $T_e = 50$ and 60 ms.

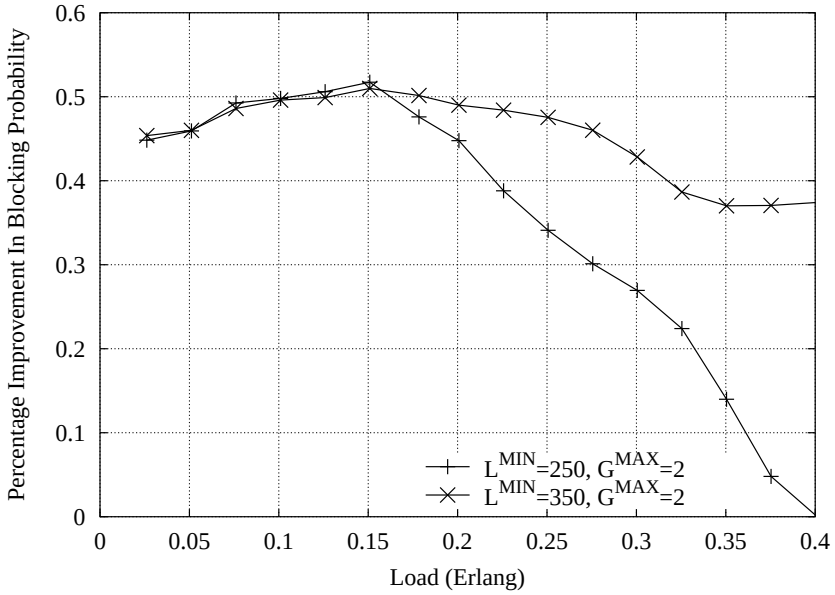


Figure 6.20. The percentage improvement in packet blocking probability of NoRO compared to no grooming assuming $G^{MAX} = 2$ and L^{MIN} changes from 250 to 350.

used. The results demonstrate that burst grooming provides lower blocking probability compared to the case with no grooming, when no wavelength converters are utilized. This is because burst grooming allows the incoming groomed sub-bursts, which have been re-assembled to be retransmitted on *any* available channel. Such advantage is diminished when all nodes have wavelength converters. In terms of average end-to-end packet delay, having wavelength converters appears to have no impact.

Based on the above results, it can be concluded that burst grooming can particularly be advantageous for networks which are constrained by cost (e.g., having no wavelength converters) or technology (e.g., having core nodes with slow switching times) and which carry on-demand traffic with relatively low arrival rate sub-bursts.

6.5 Conclusion

In this chapter we discussed the problem of data burst grooming in optical burst-switched networks. The main motivation for this study is improving network performance when the sub-bursts have low arrival rate, and the core node's switching time is larger than the average size of sub-bursts. Under such assumptions, sub-bursts will time out before they reach their minimum required length and hence, padding overhead must be added. We developed two grooming algorithms, namely MinTO and NoRO, which aggregate multiple small sub-bursts together in order to reduce the padding overhead, while minimizing any added routing overhead.

Through a comprehensive simulation study we investigated the performance of the MinTO and NoRO algorithms in terms of packet blocking probability and average end-to-end packet delay. Our results show that, in general, the proposed grooming algorithms can improve the performance when compared with the case with no grooming. However, careful considerations must be given to loading conditions and the number of sub-bursts allowed to be groomed together. This is due to the fact that they alter the network traffic characteristics negatively and make the traffic more sporadic.

One area of future work would be to extend the proposed burst grooming framework

such that it can support service differentiation and QoS. As we mentioned before, two potential drawbacks of burst grooming are increase in number of electrical-to-optical converter/transmitter and additional buffering requirements. Further studies are required to examine such cost increases. In addition, analyzing the cost-performance comparison between two networks, one with burst grooming capability but no wavelength converters and the other with wavelength converters but no grooming capability, can also be interesting. Another open problem to study is the data burst grooming under static traffic scenario, where the average traffic between each node pair is known in advance.

CHAPTER 7
A MULTI-LAYERED APPROACH TO OPTICAL BURST-SWITCHED BASED GRIDS

7.1 Introduction

Today, more than any other period in history, scientific and business communities, are in need of massive computational ability, data storage, and collaborations. With the astonishing advances in telecommunications and development of countless communication devices, such needs are expected to grow far beyond latest technological advances. For example, by 2015 it is estimated that particle physicists will be requiring exabytes (10^{18} bytes) of storage and petaflops per second of computation [145]. These types of requirements have motivated the researchers to develop the Grid. The Grid provides a practical and cost efficient infrastructure to accommodate scientific and business communities with their integrated computer-intensive requirements.

Over the years various Grid projects have sought ways to share available resources for a broad set of applications in science, business, environment, health, and other areas. In general, the most common resources in the Grid are computational power, data storage capacity, and networking capability [146]. The heart of the Grid, however, is its network. Adequate networking allows geographically dispersed resources to be utilized collectively in order to satisfy a given application. Clearly, resource utilization of the Grid is limited by the available link bandwidth. Hence, integrating Grid resources with emerging high-performance optical network, including optical switching and Dense Wavelength Division Multiplexing (DWDM) technologies, appears to be the natural choice [147], [148]. A number of experimental testbeds, including Optical Metro Network Initiative (OMNI) [149], CA*net4 [150], or TransLight [151], have focused on developing such ubiquitous high-performance networking blocks for the Grid.

In general, the enabling technologies in the optical network block of the Grid, including

the switching and resource allocation mechanisms, may be different depending on the Grid application. For example, a particular application may require moving a large amount of data (e.g., transferring multi-petabytes of astronomical data generated by new e-Astronomy experiment such as VISTA ([145]-Chap. 36). For such applications, efficient and dynamic reservation of *lightpaths* are required at the Grid network level to guarantee sufficient bandwidth throughout the duration of the requests [152], [157]. A lightpath is typically defined as a dedicated end-to-end optical connection between two or more optical nodes. We refer to such Grid enabling architecture as Optical Circuit Switched (OCS)-based Grid or *Grid over Optical Circuit Switching* (GoOCS).

Many other grid applications require computationally intensive resources (e.g., mathematical problems requiring large number crunching). In fact, it is conceivable to imagine a multiple number of users each with sub-wavelength bandwidth requirements but large processing power needs. In these cases, the data is transmitted to proper Grid resources and the results are sent back to the clients after the data processing is completed. Such applications are often small in size, sensitive to latency, and require guarantee of service. Hence, satisfying these applications through establishing dedicated lightpaths, which include path setup and path teardown and can take as many as tens of seconds [154], may not be efficient.

An alternative approach to meet computationally intensive Grid applications with moderate data size is to implement a new optical switching paradigm called optical burst switching (OBS) [38]. In this architecture, referred to as *Grid over Optical Burst Switching* (GoOBS), one or more application requests, or *jobs*, are assembled into a super-size packet called *data burst*, which are then transported over the optical core network and forwarded to their appropriate Grid resources. Each data burst has an associated *control packet* containing information such as burst's duration, source node, the type of the Grid resources the burst requires, etc. Typically, the control packet is separated from the burst in space and time, transmitted on dedicated channels apart from its associated burst by an offset.

An attractive feature of GoGBS is its support of existing DWDM optical networking in-

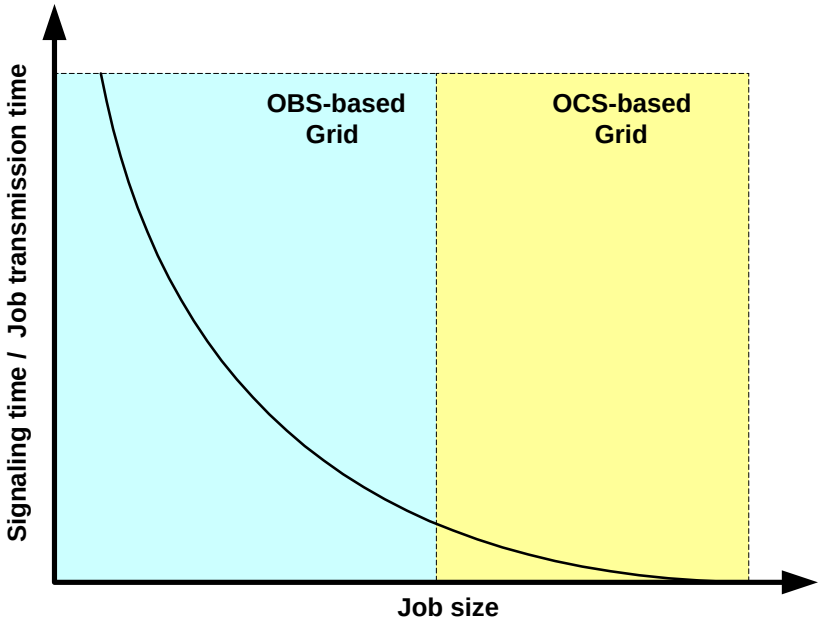


Figure 7.1. The ratio of signaling time over total transmission time of a request (job) between the client and Grid resources as the job size varies.

frastructure and minimizing the need for optical-electrical converters at intermediate nodes. Another important advantage of GoOBS is its ability to utilize link bandwidth and Grid resources efficiently and to provide low end-to-end latency. A working group in the Global Grid Forum (GGF) is committed to the standardization of OBS in the context of Grid computing [165].

Fig. 7.1 shows the ratio of the signaling time, including the time required to set up and tear down lightpaths, over request (job) transmission time between the client and Grid resources as a function of job size. As demonstrated in this figure, if the ratio is reasonably small, say 5%, it is feasible to utilize OCS-based Grid. However, as the data size reduces and applications become more latency-sensitive, OBS-based Grid tends to be more efficient.

Implementing OBS as the transport mechanism for the Grid is a relatively new area and many important issues pertaining the GoOBS architecture are still uncovered. For example, it is not well understood how to aggregate multiple jobs in a single burst, how to retransmit a job in case of data burst loss, or how to route jobs to unspecified proper Grid resources

to optimize Grid's utilization; the later issue is known as the *anycast routing problem*. A handful of works have discussed GoOBS. In [155] the authors discuss solutions towards an efficient and intelligent network infrastructure for the Grid and propose taking advantage of recent developments in optical networking technologies, including OBS. In [157] basic advantages of OBS-based Grid are mentioned and its generic architecture is discussed. An OBS-like signaling protocol, called Just-In-Time is introduced in [156] to enable optical networking for Grids.

The main contribution of this chapter is two-fold. First, we present a unique layered architecture for Grid-over-OBS. In our architectural representation, we position OBS protocol stack within the framework of the layered Grid architecture. We describe how different layers of the Grid interact with OBS layers and elaborate on protocols supported by each layer. Second, we present a generic framework for anycast routing in the context of GoOBS when jobs don't have explicit addresses and they can be serviced by any appropriate Grid resource. We develop several algorithms to support anycasting when only a single copy of a job is transmitted. Through simulation analysis, we show the performance of our anycast algorithms and compare them with the shortest-path unicast routing in which all jobs have specific addresses. This performance comparison will be based on average end-to-end delay and blocking probability of jobs.

The rest of this chapter is organized as follows. In Section 7.2, we briefly review the general layered Grid architecture. In Section 7.3, we describe how OBS protocol stack can be positioned within the layered Grid architecture. In Section 7.4, we elaborate on anycast routing problem and introduce our anycast routing algorithms. Finally, in Section 7.5 we present performance results obtained by means of simulations, followed by concluding remarks in Section 7.6.

7.2 General Grid Architecture

In this section we review the proposed layered Grid architecture [159]. At the center of the Grid foundation lies the fundamental concept of a virtual organization, i.e., a group

of individuals or institutions that interact with each other and their resources according to a set of rules, or protocols. The Global Grid Forum (GGF) has considered a layering approach to develop such protocols. The idea of layering allows for high-level functions to use common lower-level functions. The layered Grid architecture, as proposed by GGF, is shown in Fig. 7.2(a). We briefly describe each layer from bottom to up and name its basic functionalities [160].

- *Fabric:* provides the underlying base structure including the storage systems, computers, networks, system descriptors, etc.
- *Connectivity:* defines core communication and the capabilities of resources. It also defines the authentication, authorization, delegation utilities of the users. Communication protocols enable the exchange of data between Fabric layer resources and includes transport, routing, and naming.
- *Resource:* provides access to information and computation. This layer provides information about the state, performance, and structure of the grid system.
- *Collective:* deals with interactions that are global in nature, like resource discovery, system monitoring, etc. This layer also enables user application specific jobs, such as archiving, checkpointing, management, etc.
- *Application:* refers to many different commercial, scientific, engineering applications requiring one or more resources such as computing power and speed, data storage, data federation and availability, etc. provided by the Fabric layer.

Fig. 7.2(b) magnifies the Grid Resource and Connectivity protocol stack and demonstrates how the OBS technology can be positioned within the context of the Grid layer architecture. In general, as demonstrated in this figure, OBS can be considered as the networking technology at the *lower layers* of the protocol model providing alternatives for the physical, data link, and network layers. In our model, the Connectivity and Resource layers of

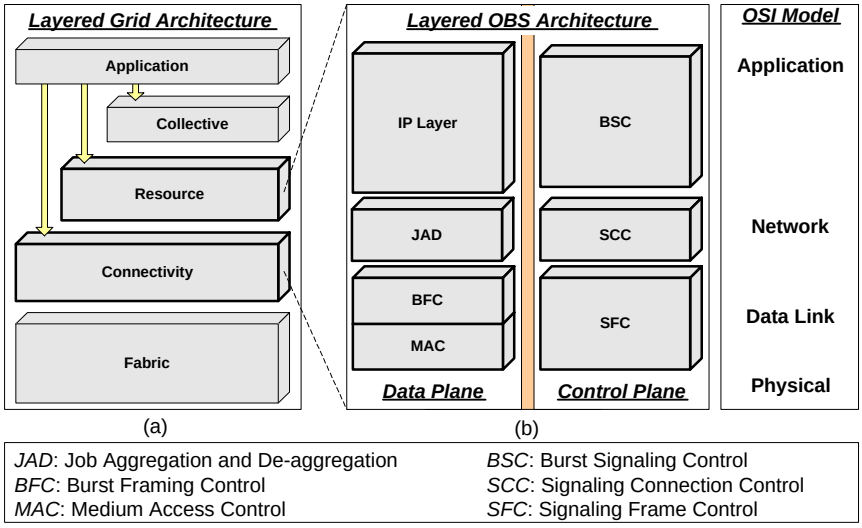


Figure 7.2. (a) A layered Grid architecture; (b) layered Grid-over-OBS architecture.

the Grid architecture act as the application layer of OBS protocol. Although other alternatives maybe considered, without loss of generality, in this section we mainly focus on a IP-centric Grid, shown in Fig. 7.3(a), where the communication protocols in the Connectivity layer of the Grid are based on the TCP/IP protocol stack. In other words, all jobs requiring grid resources are framed as IP packets. A specific example of the Connectivity layer supported by current Globus Toolkit is shown in Fig. 7.3(b) [161].[1]

7.3 Grid-Over-OBS Architecture (GoOBS)

We now briefly describe the OBS layer architecture, shown in Fig. 7.2(b), functioning as the networking layer of the Grid. Our OBS layered representation closely follows the OSI reference model. In our representation, we separate the control plane functionalities and protocols from those of the data plane [158].[2] Such separation appears natural since the

[1]The Globus Toolkit is an example of Open Grid Services Architecture (OGSA) maintained by the Globus Alliance and it grid-enables a wide range of computing environments. It is a software tool kit addressing key technical issues in the development of grid-enable environments, services, and applications and has widely been adapted as a grid technology solution for scientific and technical computing.

[2]In Chapter 3.1, we presented the OBS network architecture in a layered manner as a set of protocols that can provide various services and exchange data with one other. For more information on details of each sublayer in data and control plain refer to Chapter 3.1.

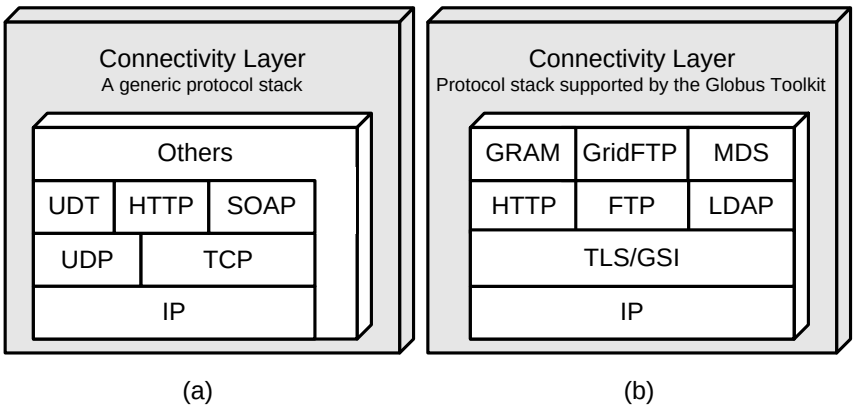


Figure 7.3. (a) An example of the communication protocol stack provided by the connectivity layer in the Grid; (b) an example of communication protocol stack supported by current Globus Toolkit.

control information is transmitted out-of-band in OBS networks.

7.3.1 OBS data plane

The data plane transports incoming jobs from the higher Grid layers to proper Grid resources and ensures that the job has properly been processed within the appropriate requested time, called the *job slack time*.

Job Aggregation and De-aggregation (JAD) Layer: The JAD layer aggregates incoming jobs with similar properties, such as quality-of-service or type of Grid resources required. In general, all jobs entering the JAD layer can be divided into two main categories: *unprocessed* and *processed* jobs. Bursts aggregating unprocessed jobs with similar properties are called *roving bursts*, because such bursts typically have no explicit destination address. On the other hand, bursts aggregating processed jobs returning to the same single or multiple destination clients are called *destined bursts*. In this case, JAD translates the client address into an OBS equivalent node address. For roving bursts, the JAD determines the OBS nodes, which can support the required Grid resources. Such information can be provided by the upper Grid layers, namely, the Resource and Connectivity layers.

The JAD layer also de-aggregating the received data bursts. For example, when destined bursts are received, all embedded processed jobs are extracted by JAD and sent to

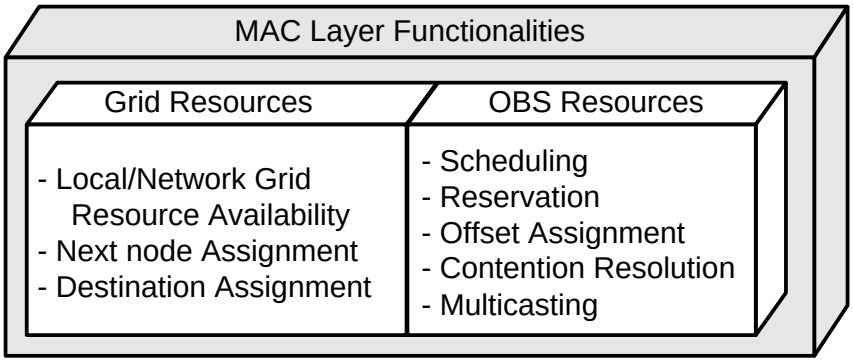


Figure 7.4. OBS MAC layer functionalities.

appropriate destination clients.

Burst Framing Control (BFC): The function of the burst framing control layer is to receive the aggregated jobs embedded in a burst from the higher layer (JAD) and to encapsulate them into proper frame structures. This layer also decodes incoming data burst frames and extracts its processed or unprocessed jobs.

Medium Access Control (MAC): The MAC sublayer in data plane maintains two types of information, as shown in Fig. 7.4, namely, OBS resources and Grid resources.

Protocols pertaining to OBS resources include reservation and scheduling protocols, offset time assignment protocols, contention resolution schemes, and multicasting protocols. Examples of out-of-band one-way reservation protocols are JIT, and JET, which have been proposed for OBS networks [45]. Some of the common scheduling algorithms considered for OBS include latest available unscheduled channel (LAUC) or Horizon Scheduling, and latest available unscheduled channel with void filling (LAUC-VF) [39] and [100]. An important function of the MAC layer is to resolve contention and reduce burst loss. Burst loss occurs due to lack of either Grid resources or sufficient OBS network resources, including output port or available grid resources.

The MAC layer also maintains a list of available local Grid resources. Using such information, the source node determines which nodes can process the embedded jobs in the burst. Upon receiving a request by an incoming roving burst, the MAC layer decides

whether to process the burst or forward it to the next node.

7.3.2 OBS Control plane

We now turn our attention from the data plane to the control plane. In our layered OBS model the MAC sublayer operates as *application layer* of the control plane allowing scheduling and reservation protocols to be performed in a domain (electrical) independent of data (optical).

Burst Signaling Control (BSC): The BSC layer receives the data burst properties, including Grid resource type, destination address, quality-of-service, burst type (roving or destined), etc., from the MAC and determines the type of *control packet* to be transmitted to the next hop. Typical examples of the control packet types are burst header packets (BHP), network management packets (NMP), and burst confirmation packets (BCP). BHPs contain their associated data burst properties. NMPs provide network status information including congestion status and possibly available Grid resources of each OBS node. BCPs can be used to confirm that a burst has found the proper Grid resources and it is expected to be processed and returned within some time units.

Signaling Connection Control (SCC): The SCC layer includes the routing algorithms for control packets in order to establish the physical path for outgoing data bursts. The actual data burst routing also takes place in this layer.

In the context of GoOBS, various routing protocols can be considered for implementation in the SCC layer. Such routing protocols are *unicast*, *multicast*, *anycast*, or a combination of them depending on the data burst type. Anycast routing protocols are implemented when no explicit destination nodes has been assigned to roving bursts. We will introduce a number of anycast protocols in the next section, which can be utilized in GoOBS.

Signaling Frame Control (SFC): The SFC layer receives bit streams containing the control packet type and its associated data burst properties, and it constructs control packet frames by attaching overhead bits. Table below lists possible fields in a BHP frame associated with a roving or destined bursts.

Table 7.1. Control Packet Frame Fields

BHP Frame Field	Description
Type	Burst type; can be Destined or Roving, indicating its associated burst is carrying processed or unprocessed jobs, respectively
Id	Burst identification used for job sequencing
Ingress Channel	Wavelength channel carrying the data burst
Duration	Duration of the data burst in units of time
Offset	Offset between data burst and its associated control packet in unites of time
Destination	Destination OBS core node where sufficient Grid resources are available.
Routing History	Includes information such as the number of physical hops the data burst has passed through, which nodes it has visited, etc.
Processing Period	The amount of processing time either required or consumed for the jobs embedded in bursts
Stack Time	The maximum end-to-end time delay before the burst is expired
O&M	Related to network management signaling information such as loop-back requests, protection switching, or link failure notification, etc.

7.4 Anycasting Routing Protocols in GoOBS

In this section we first describe our basic network assumptions in GoOBS and compare fundamental differences between OBS-based Grid and traditional OBS networks. Then, we formally introduce the general anycast routing problem and algorithms supporting it.

7.4.1 Network assumptions

A generic network architecture of GoOBS, including DWDM links, Grid edge nodes and its interfaces to Grid resources, and OBS core nodes is provided in [157]. We consider the following network assumptions: the network consists of $|N|$ nodes and $|L|$ links, represented by sets $N = \{1, 2, 3, ..., n\}$ and $L = \{(1, 2), ..., (j, k)\}$, respectively; each burst has a maximum tolerable end-to-end delay (stack time, T_{slack}) upon processing a roving burst, a confirmation control packet is sent back to the source to notify where the jobs are being processed; and processed jobs always have higher priority, hence, destined bursts can preempt roving bursts.

An OBS-based Grid is fundamentally different from the traditional IP-centric OBS network in a number of ways. For example, in GoOBS, jobs embedded in a burst must be *returned* to their original source nodes (clients). In addition, a burst can be discarded for (at least) *two* reasons: burst contention at the intermediate node *and* lack of sufficient Grid resources throughout the network within a predetermined time period (stack time). We refer to the later as *burst starvation*.

Another fundamental difference is that unlike IP-centric OBS networks, jobs in the Grid may be assigned no explicit destination address, as long as they are properly processed and returned to their clients. Consequently, instead of requiring shortest-path-based unicast routing protocols to transmit a burst to a specific destination node, GoOBS supports *deflection-based anycast* protocols as its underlying communication mechanism. In such protocols, a burst can be sent to any OBS node with appropriate Grid resources, and intermediate nodes that lack sufficient resources simply *deflect* the burst to the next proper hop.

7.4.2 Problem formulation

IP-based anycasting has been considered and discussed in many works, including [162], [163], and [164]. Using the same basic concept, we define anycasting in the context of GoOBS as follows: a client transmits a job to an anycast address, and the OBS network is responsible for providing best effort delivery of the job to at least one, and preferably only one, of the proper Grid resources accepting the anycast address. It is, evident that strictly speaking, unicasting or multicasting are both special cases of anycasting.

The following formulation can be derived for GoOBS anycasting. Assuming the entire GoOBS network (including the physical topology, full routing knowledge, and all available Grid resources associated with each core node) is known; *given* a burst $B(s, r)$ with source node s and required Grid resources provided by set of nodes $r \subseteq D = \{r_1, r_2, ..., r_d\}$, where $s \in N$, $D \subseteq N$, and $B(s, r_1), B(s, r_2)$, etc. are identical copies of $B(s, r)$; *find* the minimum size $|r|$, $|r| \geq 1$, such that the blocking probability of $B(s, r)$ is minimized, *subject* to burst's slack time. We represent the original burst as $B(s, r_1)$.

Upon constraining the $|r|$, called *anycasting grouping size*, two different categories of anycasting protocols can be considered:

- *Single-copy anycasting, ($|r| = 1$):* A single copy of the job request (embedded in a single data burst) is transmitted by the source;
- *Multiple-copy anycasting, ($|r| > 1$):* Multiple copies of the job request (multiple data bursts) are sent to multiple nodes with proper resources.

We note that multiple copies of a burst can be generated at the source or at one or more intermediate nodes where bursts containing the job are cloned [97]. Clearly, when multiple number of bursts are generated, care must be taken to avoid looping and unintentional processing of the same burst by multiple nodes. In this section, we only focus on single-copy anycasting, where $|r| = 1$, and propose a number of heuristic algorithms to implement it.

7.4.3 Anycasting Algorithm Description

In general, single-copy anycast routing consists of two basic operations: *destination assignment* and *burst deflection*. The detail functionalities of each operation varies depending on burst type and whether nodes are stateful or stateless, that is if network status is communicated between nodes or not, respectively.

We consider three distinct burst destination assignment schemes performed by the source node.

- *Soft assignment (SA):* The source selects a destination node with available Grid resources for the burst. This selection can be random or according to some weighted function. The assigned *soft* destination can be altered by other nodes due to contention or starvation. In addition, an intermediate node can accept a burst with a different soft destination, if the node has sufficient processing resources.

- *Hard assignment (HA):* This is similar to SA, however, the assigned destination node by the source *cannot* be altered by any intermediate node. Note that HA is basically unicasting, which is considered to be a special case of anycasting.

- *No assignment (NA):* The source assigns no explicit destination node for the outgoing roving burst and it just post the burst (containing one or more jobs), hoping the burst finds the adequate grid resources and jobs are processed. Therefore, the burst will wander in the network until it finds appropriate grid resources. If the stack time of the burst is expired, the burst will be discarded. When the burst arrives at an intermediate node, the node checks its available resources and if sufficient resources were not available the burst is forwarded to the next selected hop.

Burst deflection operation can be triggered at an intermediate or destination node due to contention or lack of sufficient processing resources, respectively. The way in which burst deflection is implemented varies depending on the burst type.

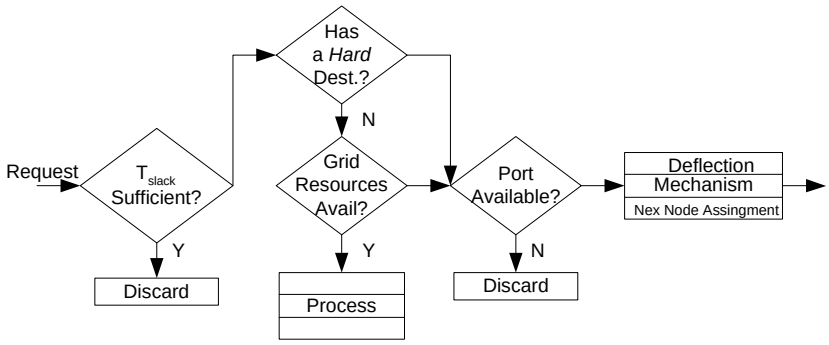


Figure 7.5. Basic steps in burst deflection operation.

Fig. 7.5 abstracts the general treatment of an incoming roving burst by an intermediate node. Note that the burst is initially checked for its slack time, T_{slack}, to ensure the burst is valid:

$$T_{slack} \geq T_{tx} + T_{proc} + T_{agg}. \tag{7.1}$$

In the above expression, T_{tx} is the transmission delay from the OBS node to the client, T_{proc} is the estimated required processing time, and T_{agg} is the re-aggregation time as the job processing is completed and the processed job is ready to be returned to the client.

An important issue in burst deflection in case of contention is determining *where* to deflect the burst to. We consider three burst deflection schemes according to different resource availability criteria:

- *Random port availability (RPD):* In this case, upon contention, the burst is deflected to an available randomly selected egress port. This scheme is similar to the hot-potato protocol in the sense that the node forwards the burst to the first available channel on any randomly selected egress port.

- *Weighted port availability (WPD):* This is very similar to RPD, except the port selection at the node is based on some weighted function. Such function can include, for example, the port's blocking probability, whether the port is on an alternative shortest path to the original destination, etc.

- *Weighted Grid-resource availability (WGD):* In this case the node examines all available grid resources throughout the network. Then, according to a weighed function, the node decides which egress port should be selected in order to forward the contending burst. The weight function can be *shifted* in favor of the ports providing alternative shortest paths to the original destination node.

Using the above framework, we consider a number of algorithms and describe their details below. We emphasize that our motivation in selecting these algorithms is to focus on anycasting and to compare its performance with the traditional shortest-path-based unicast algorithms.

Soft destination assignment with no deflection (SA-ND): In this case a randomly selected destination is assigned to each outgoing burst, and the burst will be routed on its shortest path toward the assigned destination. However, the burst can be processes by the first node with available resources along the shortest path. If the burst reaches its destination node and no processing resources were available, a new soft destination will be assigned.

Hard destination assignment with no deflection (HA-ND): In this case each burst has a randomly assigned destination, and it is forwarded along the shortest path to the assigned destination. If the assigned destination does not have sufficient resources to process the jobs embedded in the burst, a new destination will be assigned to the burst. Note that HA-ND is equivalent to the traditional shortest-path-based unicast routing algorithm.

No destination assignment with no deflection (NA-ND): In this case we assume that no burst has an assigned destination, as in NA. Upon arrival at intermediate nodes, the burst is randomly assigned to an egress port which may or may not be available. If the selected port is not available, the burst will be dropped. The motivation for studying this algorithm is two-fold: to ensure that the load is properly balanced throughout all the egress ports at each node; and to use NA-ND as a baseline to study other variations of anycasting algorithms where no explicit destinations are assigned.

Depending on the deflection mechanism, we consider three different variations of the ND anycasting algorithm:

- *No destination assignment with random port deflection (NA-RPD):* This is similar to NA-ND. However, in case the first randomly selected egress port was not available, the burst can be deflected to another randomly selected egress port on the node. The selection will continue until an available port is found. If no such port was found, the burst will be discarded.

- *No destination assignment with weighted port deflection (NA-WPD):* In this case, if the first selected egress port is busy, the node will assign an alternative egress port. The port selection is based on finding the least congested egress port with the lowest measure blocking probability.

- *No destination assignment with weighted Grid-resource availability deflection (NA-WGD):* In this case, when contention occurs at node i and the first selected egress port is no longer available, the node must find an alternative egress port. This is performed by calculating the *Grid-resource availability function*, Γ_p, for each remaining port p:

$$\Gamma_p = \Sigma_{j,j\neq i} \frac{\Omega_j}{H_p(i,j)}, \tag{7.2}$$

  In this expression, Ω_j are the available Grid-resource of node j, which have *not* been visited by the contending burst; $H_p(i,j)$ is the shortest-path from node j to node i through port p. If there is no path between node pair (i,j), or such a path is not the shortest-path through port p, $H_p(i,j)$ will be set to infinity. Using the above function, the alternative port will be the one with the largest Γ value.

7.5 Performance Results

In this section we present the simulation results obtained by implementing the aforementioned algorithms. We consider the European core network, containing 13 nodes and 17 bidirectional links, as our test network. We assume all ports have 4 wavelengths each operating at 40 Gbps. Furthermore, we assume that all jobs can be processed by all nodes as

long as the nodes have available Grid resources. We implemented JET [45] as the wavelength reservation technology. We assumed Poisson job arrivals at each client. We focus on two performance metrics as the network load (in Erlang) varies: the job blocking probability and average job hop count.

Fig. 7.6 compares the blocking probability of bursts (jobs) obtained for no-destination-assignment, soft-destination-assignment and hard-destination-assignment when no deflection is implemented, denoted by NA-ND, SA-ND, and HA-ND, respectively. An interesting observation in this figure is that the performance of SA-ND is much better than HA-ND when hard destinations are assigned to bursts. Another interesting observation is that, at higher loads, when no destinations are assigned to bursts, lower job blocking probability can be achieved compared to when bursts are given specific destinations.

Fig. 7.7 shows the average hop count obtained by implementing NA-ND, SA-ND, and HA-ND. Note that the lowest average hop count is achieved by NA, which is considerably lower than the case when bursts are given specific destination nodes. This is because in NA, jobs tend to be processed by the nodes close to the source.

Next, we examine the performance of the NA anycasting algorithm. Fig. 7.8 shows the blocking probability of NA using different deflection mechanisms, namely no-deflection, random-port-deflection, weighted-port-deflection, and weighted-Grid-resource-based deflection, denoted by NA-ND, NA-RPD, NA-WPD, and NA-WGD, respectively. Our results indicates that NA-WGD results in the lowest blocking probability. Using the Grid-resource function, NA-WGD can utilize resources more efficiently. Fig. 7.8 indicates that the performance of NA-RPD and NA-WPD is very similar. This is in spite of the fact that NA-WPD is more complex in terms of hardware implementation because it requires maintaining port statistics.

A drawback of deflection is an increase in the average hop count, as shown in Fig. 7.9. Note that NA-WGD appears to be a good tradeoff between job blocking and average hop count.

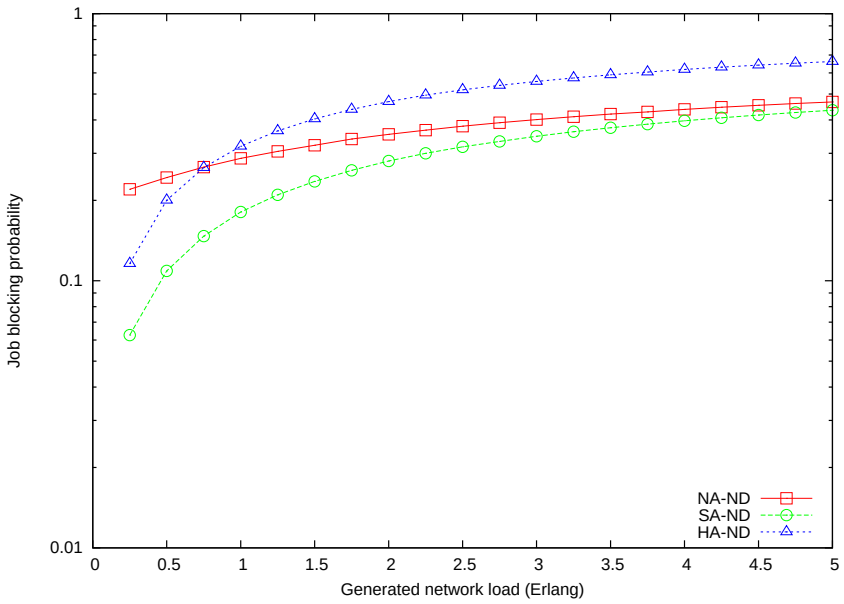


Figure 7.6. Job's blocking probability when no deflection is implemented.

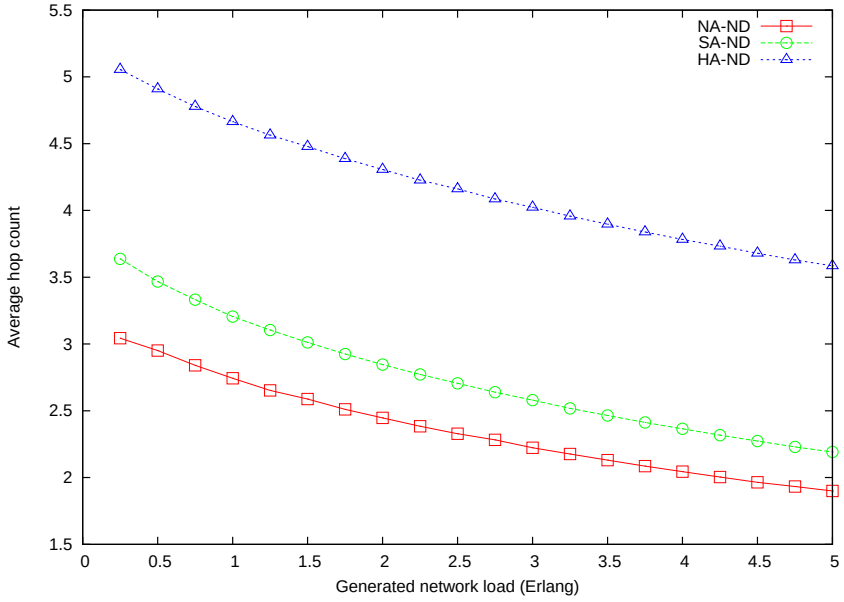


Figure 7.7. Job's average hop count when no deflection is implemented.

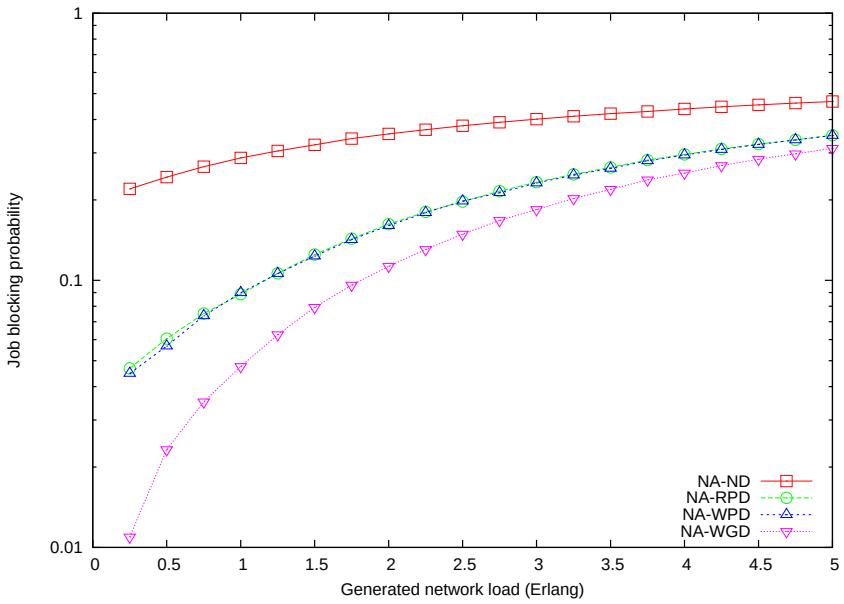


Figure 7.8. Job's blocking probability for no-destination assignment using different deflection mechanisms.

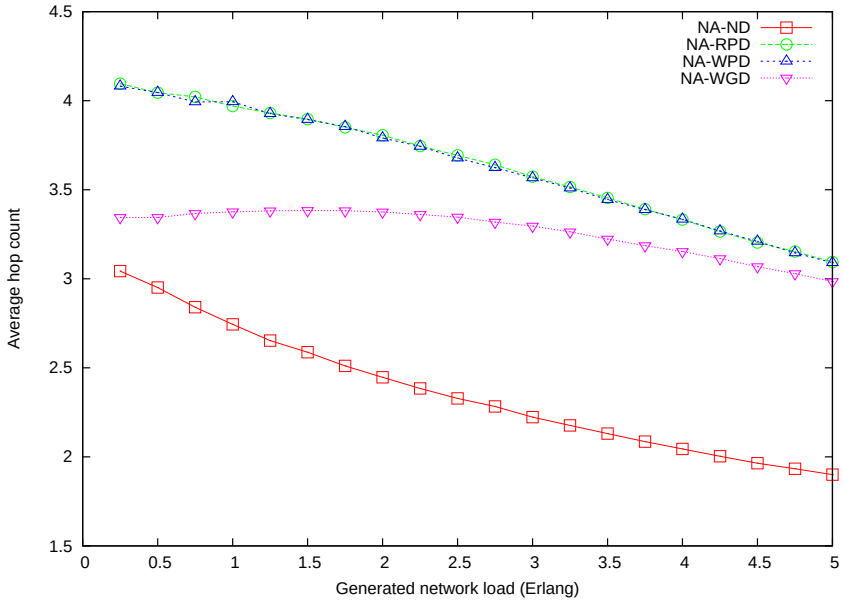


Figure 7.9. Job's average hop count for no-destination assignment using different deflection mechanisms.

7.6 Conclusion

In this chapter we presented a formal definition of layered OBS network which can be positioned to support the Grid architecture. We referred to such architecture as Grid-over-OBS. We showed the potential gains using OBS-based Grid architecture and presented a generic framework for anycasting routing in the context of Grid-over-OBS. We developed several anycasting algorithms, and, through computer simulation, we examined the performance of each and compared them with the traditional unicasting. Our results indicate that, in general, when jobs are allowed to be processed by any node with available resource, lower blocking probability can be obtained. Furthermore, blocking probability can be improved by implementing more sophisticated deflection mechanisms.

CHAPTER 8
CONCLUSION

The amount of raw bandwidth available on fiber optic links has increased dramatically with advances in dense wavelength division multiplexing (DWDM) technology; however, existing optical network architectures are unable to fully utilize this bandwidth to support highly dynamic and bursty traffic. As the amount of bursty Internet traffic continues to grow, it will become increasingly critical to develop new architectures to provide the flexible and dynamic bandwidth allocation to support this traffic.

Optical burst switching (OBS) was proposed as a new paradigm to achieve a practical balance between coarse-grained circuit switching and fine-grained packet switching. OBS provides dynamic bandwidth allocation and statistical multiplexing of data, while having fewer technological restrictions than optical packet switching. In recent years, numerous studies have been dedicated to address various issues and challenges in OBS technology.

In this dissertation we provided several architectures and protocols for solving some of the fundamental challenges facing the optical burst-switched networks. In this chapter, we will summarize the contributions of this work and provide some directions for future research.

8.1 Summary of Research Contributions

In Chapter 1, we outlined the basic properties of optical burst switching technology and how it is compared with other optical switching technologies.

In Chapter 2, we provided an architectural overview of edge and core nodes in OBS networks. We also provided a brief survey of issues pertaining OBS network, such as burst assembly, quality-of-service, contention resolution, and supporting IP over OBS.

In Chapter 3, we defined a new multi-layered architecture for supporting optical burst

switching (OBS) in an optical core network. In this architecture we considered both the control plane as well as the data plane. The functionality and the primary protocols that are required at each layer were explained and the interaction between these layers were discussed.

In Chapter 4, we introduced an effective reactive contention resolution mechanism called *look-ahead* contention resolution. We also introduced a special case of this mechanism called shortest-drop policy which is more feasible to implement in hardware. In this chapter we proposed a hardware architecture and studied the hardware feasibility of our proposed contention resolutions.

In Chapter 5, we proposed a feedback-based proactive contention resolution policy which is suitable for reducing contention in OBS networks. We showed that, in this scheme, contention avoidance is achieved by dynamically varying the data burst flows at the source to match the latest status of the network and its available resources. Thus, as the available network resources are changed, a source should vary its data burst transmission rate (or its offered load) to the network, accordingly. In our work, we compared the performance of our proposed feedback-based and other traditional reactive contention resolution mechanisms.

In Chapter 6, we introduced a new concept called *burst grooming*. We showed that by grooming multiple sub-bursts together, it is possible to minimize the amount of padding overhead generated in the network, when the IP packet arrival rate is low. We introduced an edge node architecture enabling burst grooming capability and developed two basic grooming approaches. Through a comprehensive simulation study we showed that, in general, our proposed grooming algorithms can improve the performance compared to the case of no grooming.

In Chapter 7, we examined the basic concept of *Grid-over-OBS*. We presented a formal definition of layered OBS network which can be positioned to support Grid architecture. We showed the potential gains using OBS-based Grid architecture and presented a generic framework for anycasting routing in the context of Grid-over-OBS. We developed several

anycasting algorithms and through computer simulation we examined the performance of each and compared them with the traditional unicasting.

8.2 Future Work

This section describes the extension to the work introduced in this dissertation.

In Chapter 4, we introduced a new contention resolution algorithms for optical burst switching networks called Look-ahead Contention Resolution (LCR). We discussed the algorithm details as well as its implementation complexity and examined its performance in terms of burst loss probability for different classes of service. We also presented a generic hardware architecture for fast BHP processing and discussed the design details of the BHP scheduler unit.

One area of future work would be to extend the proposed look-ahead contention resolution to include limited buffering. Examining the fairness of the algorithm is also another important issue. For example, it is interesting to investigate whether LCR tends to favor longer and drop shorter bursts or, on average, treat all bursts similarly. Furthermore, we intend to use our proposed general hardware architecture for the scheduler unit such that it can replace the conventional event driven computer simulation. Under hardware simulation testbed a much deeper insight into the performance of the proposed scheduling and contention resolution algorithms can be achieved.

In Chapter 5, we proposed a rate-based contention avoidance mechanism for optical burst switching networks. We demonstrated that our proposed scheme, the proportional control algorithm with explicit reduction request (PCwER), significantly reduces the packet loss probability in the OBS network. The basic trade-off of PCwER is, however, the overall reduction of network utilization due to invoking admission control when the network is congested.

An interesting study will be examining the proposed PCwER framework such that it can support service differentiation and QoS. Another important issue, which we did not

examine in this chapter is to look at burst overlapping at the edge node and find a correlation between the burst rate reduction request and the overlapping factor.

In Chapter 6, we considered the problem of data burst grooming in optical burst-switched networks. The main motivation for this study is improving network performance when the sub-bursts have low arrival rate, and the core node's switching time is larger than the average size of sub-bursts. Under such assumptions, sub-bursts will time out before they reach their minimum required length and hence, padding overhead must be added. We developed two grooming algorithms, namely MinTO and NoRO, which aggregate multiple small sub-bursts together in order to reduce the padding overhead, while minimizing any added routing overhead.

Studying burst grooming framework such that it can support service differentiation and QoS will be an interesting topic to investigate. Two potential drawbacks of burst grooming are the increase in the number of electrical-to-optical converter/transmitter and additional buffering requirements. Further studies are required to examine such cost increases. In addition, analyzing the cost-performance comparison between two networks, one with burst grooming capability but no wavelength converters and the other with wavelength converters but no grooming capability, can also be interesting. Another open problem to study is the data burst grooming under static traffic scenario, where the average traffic between each node pair is known in advance.

In Chapter 7, we presented a formal definition of layered OBS network which can be positioned to support Grid architecture. We referred to such architecture as Grid-over-OBS. We developed several anycasting algorithms and through computer simulation we examined the performance of each and compared them with the traditional unicasting. However, more efficient anycasting algorithms can be developed in OBS networks. Another interesting area of study is to model the anycasting problem in OBS. Supporting the Grid-based architecture is only one example of OBS application. More detailed studies are required to exam other types of applications, which can efficiently be supported by OBS-based networks.

The basic concept of burst switching is not limited to the optical domain. Many of the basic aforementioned techniques and models developed for OBS network, can also be extended to sensor and satellite networks. For example, sensor networks can potentially benefit from similar assembly strategies and grooming techniques developed for OBS networks. In satellite communications, where the network is less delay sensitive and has limited number of satellite switch nodes, data packets transmitted between transponders can be aggregated into data bursts with out-of-band signaling. Such networks can be more flexible and efficient than traditional SS/TDMA-based (Satellite-Switched Time Division Multiple) networks in terms of offering wide-band capacity. Many of the contention resolution policies, scheduling algorithms, as well as QoS models, specifically developed for OBS networks can be potentially utilized for burst-based satellite networks.

REFERENCES

[1] WorldCom announces 300 million expansion of UUNET network, WorldCom press release, February 19, 1997. Available at http://global.mci.com/news/releases/1997/.

[2] R. Hundt, *You Say You Want a Revolution*, Yale University Press, 2000.

[3] A. M. Odlyzko, "Internet traffic growth: Sources and implications," *Proc. SPIE*, vol. 5247, pp. 1-15, 2003.

[4] S. Romero, *Once-Bright Future of Optical Fiber Dims*, NY Times, June 18, 2001.

[5] B. Hirosaki, K. Emura, S. Hayano, and H. Tsutsumi, "Next-generation optical networks as a value creation platform," *IEEE Communications Magazine*, vol. 41, no. 9, September 2003.

[6] A. M. Odlyzko, "The myth of Internet time," *Technology Review*, 104(3), pp. 92-93A, April 2001. Available at http://www.dtc.umn.edu/ odlyzko/doc/research.html

[7] IDC (International Data Corporation) quarterly report, 2003. Available at www.idc.com.

[8] "Worldwide mobile phone market leaders: Nokia, Motorola, Samsung," IT Facts-January 28, 2005. Acailable at http://blogs.zdnet.com/

[9] W. McAuliffe, "One billion users will drive the Internet by 2005," ZDNet Online Mazazine, UK, May 30, 2001.

[10] "Asia Leads Worldwide Wireless Subscriber Base Back to Strong Growth," In-Stat June 6, 2005. Available on http://www.tekrati.com/

[11] "Effective Church Websites For Emerging Generations," Available at http://www.strategicdigitaloutreach.com/index.php/resources/churchwebsites

[12] F. Berman, G. Fox, and T. Hey, *Grid Computing: Making the Global Infrastructure a Reality*, John Wiley & Sons, 2002.

[13] D. Montgomery, M. Tacca, I. Cerutti, L. Valcarenghi, and R. Brooks, "CAD Tools in Optical Network Design," *Optical Networks Magazine*, vol. 1. no. 2, pp. 59-74, April 2000.

[14] P. Green, "Progress in Optical Networking," *IEEE Communications Magazine*, no. 54, January 2001.

[15] G. H. Sasaki and O. Gerstel, "Minimal cost WDM SONET rings that guarantee no blocking," *Optical Networks Magazine*, vol. 1, no. 4, October 2000.

[16] Gerald P. Ryan, "Setting a new Standards for Bandwidth," The Applied Technologies Group. Available at http://www.itpapers.com//techguide//dwave.pdf

[17] C. DeCusatis, "Dense wavelength division multiplexing for parallel sysplex and metropolitan/storage area networks," *Optical Networks*, vol. 2, no. 1, pp. 69-80, January/February 2001.

[18] J. F. Labourdette, "The Interconnect Penalty of Small Size Switches (Optical Networks Magazine," *Optical Networks*, September/October 2002.

[19] W. Goralski, *Sonet: A Guide to Synchronous Optical Network*, McGraw-Hill, 1997.

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

[21] F. Farahmand, X. Huang, and J. P. Jue, "Efficient Online Traffic Grooming Algorithms in WDM Mesh Networks with Drop-and-Continue Node Architecture," , *in Proceedings, Broadnets 2004*, San Jose, CA, 2004.

[22] G. Huiban, S. Perennes, and M. Syska, "Traffic grooming in WDM networks with multi-layer switches," *in Proceedings, IEEE International Conference on Communications (ICC)* 2002, vol. 5, pp. 2896- 2901, Anchorage, AK, May 2002.

[23] F. Farahmand, “Optical Technology: Recent Advances,” The University of Texas at Dallas, Department of Electrical Engineering and Computer Science, Technical Report 2005-01, January 2005.

[24] R. Clavero, F. Ramos, J. M. Martinez, and J. Marti, “ All-optical flip-flop based on a single SOA-MZI,” *IEEE Photonics Technology Letters*, vol. 17, no. 4, pp. 843-845, April 2005.

[25] P. C. Ku, C. J. Chang-Hasnain, and S. L. Chuang, “A Proposal of Variable Semiconductor All-Optical Buffer,” *Electronics Letters*, vol. 38, no. 24, November 2002.

[26] C. J. Chang-Hasnain, Pei-cheng Ku, Jungho Kim, and Shun-lien Chuang, “Variable optical buffer using slow light in semiconductor nanostructures,” in Proceedings of the IEEE, vol.91, no.11, pp. 1884-1897, November 2003.

[27] M. Lukin, “The Story Behind Stored Light: Trapping and Manipulating Photon States with Atoms,” Physics Department, Harvard University, 2003.

[28] C. Liu, Z. Dutton, C. H. Behroozi, and L. V. Hau, “Observation of coherent optical information storage in an atomic medium using halted light pulses,” Nature, no. 409, pp. 490-493, 2001.

[29] EURONANO, http://www.euronano-optic.com/euronano.htm

[30] P. R. Prucnal, “Optically processed self-routing synchronization, and contention resolution for 1D and 2D photonic switching architectures,” *IEEE Journal of Quantum Electron*, vol. 29, pp. 600-612, February 1993.

[31] S. Yao, B. Mukherjee, and S. Dixit, “Advances in Photonic Packet Switching: an Overview,” *IEEE Communications Magazine*, vol. 38, no. 2, February 2000.

[32] S. Clavenna, “40 Gig Forecast: Introduction,” Light Reading Online May 2001. Availabel at www.lightreading.com.

[33] M. Kumagai, S. Nojima, and H. Tomonaga, “IP Router for Next-Generation Network,” *FUJITSU Sci. Tech. Journal*, no. 37, pp. 31-41. June 2001.

[34] 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.

[35] D. Clark, W. Lehr, and I. Liu, “Provisioning for bursty Internet traffic: implications for industry and Internet structure,” Workshop on Internet Service Quality Economics, MIT, December 1999.

[36] D. K. Hunter, et al., “WASPNET: A Wavelength Switched Packet Network,” *IEEE Communications Magazine*, pp. 120-129, March 1999.

[37] T. S. El-Bawab and J. D. Shin, “Optical packet switching in core networks: between vision and reality,” *IEEE Communications Magazine*, pp. 60-65, September 2002.

[38] 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.

[39] J. S. Turner, “Terabit Burst Switching,” *IEEE Journal High Speed Networks*,” vol. 8, no. 1, pp. 3-16, 1999.

[40] S. R. Amstutz, “Burst Switching: An Introduction,” *IEEE Communications Magazine*, vol. 21, pp. 36-42, November 1983.

[41] S. R. Amstutz, ”Burst switching-An update,” *IEEE Communications Magazine*, pp. 50-57, September 1989.

[42] J. Kulzer and W. Montgomery. “Statistical switching architectures for future services,” *presented at ISS84*, Florence, Session 43A, May 7-11, 1984.

[43] E. Van Breusegem, J. Cheyns, B. Lannoo, A. Ackaert, M. Pickavet, and P. Demeester, “Implications of Using Offsets in All-Optical Switched Networks,” Departement of Information Technology (INTEC), University of Ghent, Belgium (online).

[44] T. Khattab, A. Mohamed, A. Kaheel, and H. Alnuweiri, "Optical Packet Switching with Packet Aggregation," *IEEE International Conference on Software, Telecommunications, and Computer Networks (SOFTCOM)*, 2002.

[45] M. Yoo and C. Qiao, "Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks," *IEEE/LEOS Conf. on Technologies for a Global Information Infrastructure*, pp. 26-27, August 1997.

[46] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood Publishers, second edition, 1993.

[47] S. Yao. S. J. Ben Yoo, and B. Mukherjee, "A Comparison Study between Slotted and Unslotted All-optical Packet-Switched Network with Priority-Based Routing," *in Proceedings, Optical Fiber Communication Conference and Exhibit (OFC) 2001*, vol. 2, 2001.

[48] 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 Optical Networking and Communication Conference (OptiComm) 2002*, Boston, MA, vol. 4874, pp. 125-136, July-August 2002.

[49] X. Yu, Y. Chen, and C. Qiao, "A Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks," *in Proceedings, SPIE Optical Networking and Communication Conference (OptiComm) 2002*, Boston, MA, pp. 149-159, July-August 2002.

[50] B. Bostica, M. Burzio, P. Gambini, and L. Zucchelli, "Synchronization Issues in Optical Packet Switched Networks," *Photonic Networks*, G.Prati, ed., Springer-Verlag, 1997.

[51] M. Yang, S. Q. Zheng, Bhagyavati, and S. Kurkovsky, "Programmable Weighted Arbiters for constructing switch schedulers," *IEEE Workshop on High Performance Switching and Routing (HPSR) 2004*, pp. 203-206, April 2004.

[52] S. L. Danielsen, et al., "WDM Packet Switch Architectures and Analysis of the Influence of Tunable Wavelength Converters on the Performance," *Journal of Lightwave Technology*, vol. 15, no. 2, pp. 219-227, February 1997.

[53] V. Eramo and M. Listanti, "Packet Loss in a Bufferless Optical WDM Switch Employing Shared Tunable Wavelength Converters," *Journal of Lightwave Technology,* vol. 18, no. 12, pp. 1818-1833, December 2000.

[54] D. J. Blumenthal, P. R. Prucnal, and J. R. Sauer, "Photonic Packet Switches: Architectures and Experimental Implementations," *in Proceedings of the IEEE,* vol. 82, no. 11, pp. 1650-1667, November 1994.

[55] C. Minkenberg, On Packet Switch Design, Ph.D. Dissertation, Eindhoven University of Technology, The Netherlands, 2001.

[56] 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, 2001.

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

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

[59] A. Banerjee, N. Singhal, J. Zhang, D. Ghosal, C. N. Chuah, and B. Mukherjee, "A Time-Path Scheduling Problem (TPSP) for Aggregating Large Data Files from Distributed Databases using an OBS Network," *in Proceedings, IEEE International Conference on Communications (ICC-04)*, Paris, June 2004.

[60] A. Detti and M. Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks," *in Proceedings, IEEE INFOCOM,* 2002.

[61] 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.

[62] E. Kozlovski, M. Duser, I. de Miguel, and P. Bayvel, “Analysis of Burst Scheduling for Dynamic Wavelength Assignment in Optical Burst-Switched Networks,” *IEEE LEOS 2001*, vol. 1, pp. 161-162, 2001.

[63] E. Kozlovski, M. Duser, A. Zapata, and P. Bayvel, “Service differentiation in wavelength-routed optical burst-switched networks,” *in Proceedings, IEEE Optical Fiber Communications (OFC’02),* pp. 774-776, March 2002.

[64] E. van Breusegem, et al., “An OBS Architecture for Pervasive Grid Computing,” *The Third Inteernational Workshop on Optical Burst Switching,* San Jose, USA, October 25, 2004.

[65] F. Farahmand, Q. Zhang, and J. P. Jue, “Dynamic Traffic Grooming in Optical Burst-Switched Networks,” submitted to *IEEE Journal of Lightwave Technology,* December 2005. (http://www.utdallas.edu/ ffarid/)

[66] F. Farahmand, Q. Zhang, and J. P. Jue, “A closed-loop rate based contention control for optical burst switched networks,” *in Proceedings, IEEE GLOBECOM*, 2005.

[67] F. Farahmand and J. P. Jue, “Supporting QoS with Look-ahead Window Contention Resolution in Optical Burst Switched Networks,” *in Proceedings, IEEE GLOBECOM 2003*, San Francisco, CA, December 2003.

[68] F. Farahmand, V. M. Vokkarane, and J. P. Jue, “Practical Priority Contention Resolution for Slotted Optical Burst Switching Networks,” *in Proceedings, IEEE/SPIE First International Workshop on Optical Burst Switching 2003*, Dallas, TX, October 16, 2003.

[69] F. Poppe, K. Laevens, H. Michiel, and S. Molenaar, “Quality-of-service differentiation and fairness in optical burstswitched networks,” *in Proceedings, SPIE Optical Networking and Communication Conference (OptiComm),* 2002.

[70] G. Thodime, V. M. Vokkarane, and J. P. Jue, "Dynamic Congestion-Based Load Balanced Routing in Optical Burst-Switched Networks," *in Proceedings, IEEE GLOBECOM 2003*, San Francisco, CA, December 2003.

[71] H. Wen, H. Song, L. Li, and S. Wang, "Load-balancing contention resolution in LOBS based on GMPLS," *in Proceedings, Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 590-594, August 27-29, 2003.

[72] I. Baldine, G. Rouskas, H. 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, 2002.

[73] 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 Optical Networking and Communication Conference (OptiComm) 2003*, Dallas, TX, vol. 5285, pp. 336-345, October 2003.

[74] J. Xu, C. Qiao, J. Li, G. Xu, and B Hall, "Efficient channel scheduling algorithms in optical burst switched networks," *in Proceedings, IEEE INFOCOM 2003*, San Francisco, USA, March 30-April 3, 2003.

[75] K. Dolzer, "Assured horizon-An efficient framework for service differentiation in optical burst switched networks," *in Proceedings, SPIE/IEEE OPTICOMM 2002*, pp. 149-159, July 2002.

[76] L. Xu, H.G. Perros, and G. N. Rouskas, "Techniques for Optical Packet Switching and Optical Burst Switching," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 136-142, January 2001.

[77] M. H. Phung, et al., "Absolute QoS Signaling and Reservation in Optical Burst Switched Networks," *in Proceedings, IEEE GLOBECOM 2004*, Dallas, USA, November 29 -December 03 (2004).

[78] M. Jeong, C. Qiao, Y. Xiong, and H. C. Cankaya, “On a New Multicasting Approach in Optical Burst Switched Networks,” *IEEE Communications Magazine*, vol. 40, no. 11, pp. 96-103 November 2002.

[79] M. Lizuka, M. Sakuta, Y. Nishino, and I. Sasase, “A Scheduling algorithm minimizing voids generated by arriving bursts,” *in Proceedings, IEEE GLOBECOM 2002,* Taipei, Taiwan, vol. 3, pp. 2736-2740, November 17-21, 2002.

[80] 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.

[81] N. Barakat and E. H. Sargent, “Dual-Header Optical Burst Switching: A new Architecture for WDM Burst-Switched Networks”, *in Proceedings, IEEE INFOCOM 2005*, Miami, USA, March 13-17, 2005.

[82] P. Du and S. Abe, “TCP performance analysis of optical burst switching networks with a burst acknowledgment mechanism,” *in Proceedings, Communications 2004 and the 5th International Symposium on Multi-Dimensional Mobile Communications-The 2004 Joint Conference of the 10th Asia-Pacific Conference,* vol. 2, pp. 621-625, August 29-September 1, 2004.

[83] 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 2003*, San Francisco, USA, December 1-5, 2003.

[84] Q. Zhang, V. M. Vokkarane, J. P. Jue, and B. Chen, “Absolute QoS Differentiation in Optical Burst-Switched Networks,” *IEEE Journal on Selected Areas in Communications,* vol. 22, no. 9, pp. 1781-1795, November 2004.

[85] Q. Zhang, V. M. Vokkarane, Y. Wang, and J. P. Jue, "TCP over Optical Burst-Switched Networks with Optical Burst Retransmission, submitted for *IEEE Journal on Selected Areas in Communications*, December 2004.

[86] R. Jain, *FDDI Handbook: High-Speed Networking with Fiber and Other Media*, Addison-Wesley, Reading, MA, April 1994.

[87] S. Gowda, R. K Shenai, K. Sivalingam, and H. C. Cankaya, "Performance Evaluation of TCP over Optical Burst-Switched (OBS) WDM Networks," *in Proceedings, IEEE ICC 2003*, Anchorage, Alaska, USA, May 11-15, 2003.

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

[89] S. Sheeshia and C. Qiao, "Burst Grooming in Optical-Burst-Switched Networks," *in Proceedings, IEEE/SPIE First Workshop on Traffic Grooming in WDM Networks,* San Jose, USA, October 2004.

[90] 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 2000,* Dallas, USA, pp. 235-246, October 2000.

[91] S. Y. Wang, "Using TCP congestion control to improve the performances of optical burst switched networks," *in Proceedings, IEEE ICC 2003,* Anchorage, Alaska, USA, vol. 2, pp. 1438-1442, May 11-15, 2003.

[92] 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 ICC 2002,* New York, USA, vol. 5, pp. 2673-2677, April 2002.

[93] 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 2002,* Taipei, Taiwan, November 17-21, 2002.

[94] W. Liao and C. H. Loi, “Providing service differentiation for optical-burst-switched networks,” *Journal of Lightwave Technology,* vol. 22 , no. 7, pp. 1651-1660, July 2004.

[95] X. Cao, J. Li, Y. Chen, and C. Qiao, “Assembling TCP/IP Packets in Optical Burst Switched Networks,” *in Proceedings, IEEE GLOBECOM 2002,* Taipei, Taiwan, November 17-21, 2002.

[96] X. Huang, Q. She, and J. P. Jue, “Multicast with Deflection Routing in Optical Burst Switching Networks,” *in Proceedings, IEEE ICC 2005,* Korea, 2005.

[97] X. Huang, V. M. Vokkarane, and J. P. Jue, “Burst Cloning: A Proactive Scheme to Reduce Data Loss in Optical Burst-Switched Networks,” *in Proceedings, IEEE ICC 2005,* Korea, 2005.

[98] Y. Chen, C. Qiao, and X. Yu, “Optical burst switching: a new area in optical networking research,” *IEEE Network,* vol. 18 , no. 3, pp. 16-23, May-June 2004.

[99] Y. Chen, M. Hamdi, D. H. K. Tsang, and C. Qiao, “Proportional QoS over OBS networks,” *in Proceedings, IEEE GLOBECOM 2001*, San Antonio, USA, November 25-29, 2001.

[100] 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. 1838-1851, October 2000.

[101] C. Gauger, “Dimensioning of FDL Buffers for Optical Burst Switching Nodes,” *in Proceedings, Optical Network Design and Modeling (ONDM) 2002*, Torino, Italy, 2002.

[102] X. Wang, H. Morikawa, and T. Aoyama, “A Deflection Routing Protocol for Optical Bursts in WDM Networks,” *in Proceedings, Fifth Optoelectronics and Communications Conference (OECC) 2000*, Makuhari, Japan, pp. 94-95, July 2000.

[103] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, “All-optical packet switching for Metropolitan Area Networks: Opportunities and Challenges,” *IEEE Communications Magazine*, vol. 39, no. 3, pp. 142-148, March 2001.

[104] F. Callegati, H. C. Cankaya, Yijun Xiong, and M. Vandenhoute, “Design issues of optical IP routers for Internet backbone applications,” *IEEE Communications Magazine*, vol. 37, no. 12, pp. 124-128, December 1999.

[105] S. Zheng, Y. Xiong, M. Vandenhout, and H. C. Cankaya, “Hardware Design of a Channel Scheduling Algorithm for Optical Burst Switching Routers,” *in Proceedings, SPIE ITCOM 2002*, vol. 4872, pp. 199-209, 2002.

[106] M. Yoo and C. Qiao, “Supporting Multiple Classes of Services in IP over WDM Networks,” *in Proceedings, IEEE GLOBECOM (1999)*, Brazil, pp. 1023-1027, December 1999.

[107] I. M. Bomze, M. Pelillo, and V. Stix, “Approximating the maximum weight clique using replicatot dynamics, *IEEE Transaction. Neural Networks,* vol. 11, no. 6, pp. 1228-1241, 2000.

[108] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling,* Wiley-Interscience, New York, NY, Apr. 1991.

[109] M. Yoo and C. Qiao. “A New Optical Burst Switching Protocol for Supporting Quality of Service,” *in Proceedings, SPIE, All Optical Networking: Architecture, Control and Management Issues,* vol. 3531, pp. 396-405, 1998.

[110] H. Vu and M. Zukerman, “Blocking probability for priority classes in optical burst switching networks,” *IEEE Communications Letters,* vol. 6, no. 5, pp. 214-216, May 2002.

[111] S. Keshav, *An Engineering Approach to Computer Networking*, Addison-Wesley, Reading, Mass., 1997.

[112] K. Boahen, “A Throughput-on-Demand 2-D Address-Event Transmitter for Neuromorphic Chips,” *in Proceedings, The 20th Conference on Advanced Research in VLSI*, Atlanta, GA, 1999.

[113] F. Farahmand and A. Shaikh “Hardware Implementation of the Shortest Drop Policy for Slotted Optical Burst Switching Networks,” The University of Texas at Dallas, Department of Electrical Engineering and Computer Science,Technical Report 20054-02, February 2003.

[114] Refer to http://www.altera.com/products/

[115] 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.

[116] T. Ozugur, F. Farahmand, and D. Verchere, “Single-anchored soft bandwidth allocation system with deflection routing for optical burst switching,” *IEEE Workshop on High Performance Switching and Routing (HPSR) 2002*, pp. 257-261, May 2002.

[117] 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)*, New York, NY, April-May 2002.

[118] C. Hsu, T. Liu, and N. Huang, “Performance Analysis of Deflection Routing in Optical Burst-Switched Networks,” *in Proceedings, INFOCOM 2002*, pp. 66-73, June 2002.

[119] X. Wang, H. Morikawa, and T. Aoyama, “Photonic Burst deflection routing protocol for wavelength routing networks”, *SPIE Optical Networks Magazine*, vol. 3, no. 6, pp. 12-19, November-December 2002.

[120] S. Oh and M. Kang, “A Burst Assembly Algorithm in Optical Burst Switching Networks,” *in Proceedings, Optical Fiber Communication Conference and Exhibit (OFC)* 2002, pp. 771-773, March 2002.

[121] M. Elhaddad, R. Melhem, T. Znati, and D. Basak “Traffic shaping and scheduling for OBS-based IP/WDM Backbones,” *in Proceedings, SPIE Optical Networking and Communication Conference (OptiComm)* 2003, Dallas, TX, vol. 5285, pp. 336-345, October 2003.

[122] K. K. Ramakrishnan and R. Jain, “A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with Connectionless Network Layer,” *in Proceedings, ACM SIGCOMM’88*, August 1988.

[123] S. Floyd and V. JacobsonRandom, “Early Detection Gate- ways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, 1(4):397-413, August 1993.

[124] K. K. Ramakrishnan and S. Floyd, “A Proposal to Add Explicit Congestion Notifcation (ECN) to IP,” RFC 2481, January 1999.

[125] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, “Ratebased congestion control for ATM networks,” *ACM Sigcomm Comp. Commun. Rev.*, vol. 25, no. 2, pp. 60-72, Aprril 1995, special issue on ATM.

[126] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore, “The ERICA Switch Algorithm for ABR Traffc Management in ATM Networks,” *IEEE/ACM Transactions on Networking*, 8(1), pp. 87-98, February 2000.

[127] D. Bansal and H. Balakrishnan, “Binomial Congestion Control Algorithms,” *in Proceedings, IEEE INFOCOM 2001*, April 2001.

[128] S. Gorinsky, A. Kantawala, and J. Turner, “Feedback Modeling in Internet Congestion Control,” *IEEE/ACM Transactions on Networking*,
Available at http://www.arl.wustl.edu/ gorinsky/pubs.html.

[129] D. Chiu and R. Jain, “Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks,” *Journal of Computer Networks and ISDN*, vol. 17, no. 1, pp. 1-14, June 1989.

[130] P. P. Mishra and H. Kanakia, “A hop-by-hop rate-based congestion control scheme,” *in Proceedings, ACM Sigcomm*, Baltimore, August 1992.

[131] K. Ohmae, et. al., “An Effeective BECN typed Deflection Routing for Optical Burst Switdhing,” *in Proceedings, IASTED CCN* 2002, pp. 259-262, 2002.

[132] H. Tanida, et. al., “An effective BECN/CRN typed deflection routing for QoS guaranteed optical burst switching,” *IEEE GLOBECOM* 2003, no.1, pp. 2601-2606, December 2003.

[133] C. Qiao, “Labeled Optical Burst Switching for IP and WDM Integration,” *IEEE Communications Magazine*, vol. 38, no. 9, pp. 104-114, September 2000.

[134] A. Okada, “All-optical packet routing in AWG-based wavelength routing networks using an out-of-band optical label,” *in Proceedings, Optical Fiber Communication Conference and Exhibit (OFC)* 2002, Anaheim, CA, March 2002.

[135] D. Katabi, M. Handley, and C. Rohrs, “Congestion Control for High Bandwidth-Delay Product Networks,” *in Proceedings, ACM SIGCOMM 2002*, August 2002.

[136] J. C. Bolot and A. U. Shankar, “Analysis of a fluid approximation to flow control dynamics, *IEEE INFOCOM 1992*, pp. 2398-2407, Florence, Italy, May 1990.

[137] S. Hardy, “All-optical Switching Groundswell Builds,” Lightwave (http://lw.pennnet.com/), May 2000.

[138] M. Casoni, E. Luppi, and M. Merani, “Impact of Assembly Algorithms on End-to-End Performance in Optical Burst Switched Networks with Different QoS Classes,” *in Proceedings, IEEE/SPIE Third Workshop on Workshop on Optical Burst Switching (2004)*, San Jose, CA, October 2004.

[139] R. Dutta and G. N. Rouskas, “Traffic grooming in WDM networks: past and future,” *IEEE Network*, vol. 16, no. 6, pp. 46-56, November-December 2002.

[140] P. J. Lin and E. Modiano, “Traffic grooming in WDM networks,” *IEEE Communications Magazine,* vol. 39, no. 7, pp. 124-129, July 2001.

[141] O. Gerstel and R. Ramaswami, “Cost effective traffic grooming in WDM rings,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 618-630, October 2000.

[142] X. Zhang, C. Qiao, An effective and comprehensive approach to traffic grooming and wavelength assignment in SONET/WDM rings, *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 608-617, October 2000.

[143] K. Zhu and B. Mukerjee, “Traffic grooming in an optical WDM mesh network,” *IEEE Journal on Selected Areas in Communication*, vol. 20, no. 1, pp. 122-133, January 2002.

[144] Y. Arakawa, N. Yamanaka, and Iwao Sasase, “Performance of Optical Burst Switched WDM Ring network with TTFR System,” *The first IFIP Optical Networks & Technologies Conference* 2004 (OpNeTec2004), Pisa, Italy, October 2004.

[145] G. Fox, A. J. G. Hey, T. Hey, and F. Berman, *Grid computing: making the global infrastructure a reality*, John Wiley & Sons, 2003.

[146] J. Joshy and F. Craig, *Grid Computing*, Prentice Hall Ptr, 2004.

[147] J. Mambretti, J. Weinberger, J. Chen, E. Bacon, F. Yeh, D. Lillethun, B. Grossman, Y. Gu, and M. Mazzuco, ”The Photonic TeraStream: Enabling Next Generation Applications Through Intelligent Optical Networking at iGrid 2002,” *Journal of Future Computer Systems*, Elsevier Press, pp. 897-908, August 2003.

[148] R. Grossman, Y. Gu, D. Hamelberg, D. Hanley, X. Hong, J. Levera, M. Mazzucco, D. Lillethun, J. Mambrett, and J. Weinberger, “Experimental Studies Using Photonic Data Services at iGrid 2002,” *Journal of Future Computer Systems*, Elsevier Press, pp. 945-956, August 2003.

[149] Optical Metro Network Initiative (OMNI), http://www.icair.org/omninet/

[150] M. Blanchet, F. Parent, and B. St. Arnaud, “Optical BGP (OBGP): InterAS Lightpath Provisioning,” IETF Network Working Group Report, March 2001. http://search.ietf.org/internet-drafts/draft-parent-obgp-01.txt

[151] T. DeFanti, C. De Laat, J. Mambretti, K. Neggers, and B. St, Arnaud, “TransLight: A Global Scale LambdaGrid for E-Science,” *Special Issue: Blueprint for the Future of High Performance Networking, Communications of the ACM,* vol. 46, no. 11, pp. 34-41, November 2003.

[152] S. Figueira, N. Kaushik, S. Naiksatam, S. Chiappari, and N. Bhatnagar, “Advance Reservation of Lightpaths in Optical-Network Based Grids,” Workshop on Networks for Grid Applications, Cosponsered by BroadNets, 2004.

[153] H. Lee, M. Veeraraghavan, H. Li and K. P. Chong, “Lambda scheduling algorithm for file transfers on high-speed optical circuits,” *IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2004),* April 19-22, 2004. http://www.ccgrid.org/ccgrid2004.

[154] T. Lavian, D. Hoang, J. Mambretti, S. Figueira, S. Naiksatam, N. Kaushil, I. Monga, R. Durairaj, D. Cutrell, S. Merrill, H. Cohen, P. Daspit, and F. Travostino, “A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks,” Workshop on Networks for Grid Applications, Cosponsered by BroadNets, 2004.

[155] D. Simeonidou, et al., “Optical Network Infrastructure for Grid”, Grid Forum Draft, September 2003.

[156] S. R. Thorpe, D. S. Stevenson, and G. K. EdwardsUsing, “Just-in-Time to Enable Optical Networking for Grids,” Workshop on Networks for Grid Applications, Co-sponsered by BroadNets, 2004.

[157] M. De Leenheer, et al., “An OBS Architecture for Pervasive Grid Computing,” Workshop on Networks for Grid Applications, Co-sponsered by BroadNets, 2004.

[158] F. Farahmand, et al., “Burst Switching Network: A Multi-layered Approach,” submitted to *Journal of High Speed Networks*, 2005.

[159] I. Foster, “The grid: A new infrastructure for 21st century science,” *Physics Today*, 54(2), 2002.

[160] I. Foster, C. Kesselman, and S. Tuecke, “The anatomy of the grid: Enabling scalable virtual organizations,” *International Journal of High Performance Computing Applications*, 15(3), 2001.

[161] D. Minoli, *A Networking Approach to Grid Computing*, John Wiley & Sons, 2005.

[162] C. Partridge, T. Mendez, and W. Milliken, “Host Anycasting Service,” RFC1546, November 1993.

[163] D. Katabi, “The use of IP-Anycast to Construct Efficient Multicast Trees,” Master Thesis, September 1998. http://ana-www.lcs.mit.edu/dina/draft-katabi-global-anycast-00.txt

[164] D. Katabi and J. Wroclawski, “A Framework for Global IP-Anycast,” INTERNET DRAFT, January 2000. Available at http://ana-www.lcs.mit.edu/dina/draft-katabi-global-anycast-00.txt

[165] D. Simeonidou and R. Nejabati, “Grid Optical Burst Switched Networks (GOBS)”, Global Grid Forum Draft, May 2005.

VITA

Farid Farahmand was born in Tehran, Iran. He received his B.S. and M.S. degrees in electrical engineering in 1993 and 1995, respectively, from the University of Texas at Dallas. He started his professional career in 1993, when he joined Alcatel U.S.A as a hardware design engineer. In January 2000, Farid moved to Alcatel Corporate Research and was involved with the development of Terabit Optical Routers. He has also been teaching at Brookhaven College since May of 2003. Farid's research interests are in high-speed packet switching and all-optical networks focusing on architecture and protocol designs for optical burst-switched networks.