2009

Burst Scheduling, Grooming and QoS Provisioning in Optical Burst-Switched Networks

Yuqi Fan
Wright State University

Follow this and additional works at: http://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
Burst Scheduling, Grooming and QoS Provisioning in Optical Burst-Switched Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

By

YUQI FAN
M.E., Hefei University of Technology of China, 2003
B.E., Hefei University of Technology of China, 1999

2009
Wright State University
I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Yuqi Fan ENTITLED Burst Scheduling, Grooming and QoS Provisioning in Optical Burst-Switched Networks BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

__________________________
Bin Wang, Ph.D.
Dissertation Director

__________________________
Thomas A. Sudkamp, Ph.D.
Director, Computer Science and Engineering
Ph.D. Program

__________________________
Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

Committee on Final Examination:

__________________________
Bin Wang, Ph.D.

__________________________
Krishnaprasad Thirunarayan, Ph.D.

__________________________
Yong Pei, Ph.D.

__________________________
Yuqing Chen, Ph.D.

__________________________
Waleed Smari, Ph.D.
Yuqi Fan, Ph.D., Department of Computer Science and Engineering, Wright State University, 2009. Burst Scheduling, Grooming and QoS Provisioning in Optical Burst-Switched Networks.

The demand of network capacity has been increasing steadily with more users than ever connected to the Internet through broadband access and the popularity of video based applications, such as YouTube. Optical wavelength division multiplexing (WDM) networks [1] are expected to form the next-generation backbone network and to fulfill the insatiable appetite for bandwidth.

Wavelength routed WDM optical networks offer the granularity of switching at a fiber, waveband, or a wavelength level. The finest granularity offered is at a wavelength level by provisioning lightpaths for different clients/services. All-optical packet switching is still deemed technically infeasible and its competitiveness as a backbone technology is debatable. Optical burst switching (OBS) presents itself as a promising technology for bridging the gap between optical wavelength switching and optical packet switching. OBS operates at the sub-wavelength level and is designed to improve the bandwidth utilization of wavelengths by exploring statistical multiplex-
ing to deal with bursty traffic, and is therefore more resource efficient than optical wavelength switching. In OBS networks, arriving data packets (e.g., IP packets) are assembled at the ingress OBS nodes to form a data burst. A burst control packet (CP) is sent on a control channel ahead of the data burst to reserve resources and configure the switches along the route traversed by the data burst [2, 3].

In this dissertation, we will explore several important and challenging issues in OBS networks in order to improve the utilization of network resource.

To reduce the switching overhead, small bursts may be groomed to reduce resource waste and switching penalty. We have studied the per-hop burst grooming problem where bursts with the same next hop may be groomed together to minimize the number of formed larger bursts and strike a proper balance between burst grooming and grooming cost, assuming all the network nodes have the grooming capability.

In order to reduce computation overhead and processing delay incurred at the core nodes, we assume that grooming can only be performed at edge nodes and the core node can send a burst to multiple downstream links, that is, the core node has light-splitting capability. We have attempted to groom small bursts into larger bursts, and select a proper route for each large burst, such that total network resources used and/or wasted for delivering the small bursts is minimized.

Optical signal transmission quality is subject to various types of physical impairment introduced by optical fibers, switching equipment, or other network components. The signal degradation due to physical impairment may be significant enough such that the bit-error rate of received signals is unacceptably high at the destination, rendering the signal not usable. Based on earlier work, we have studied scheduling
and QoS provisioning problems in OBS networks, taking physical impairments into consideration.

In the context of the JET signaling protocol, we have studied the burst scheduling problem and proposed three effective burst scheduling algorithms in OBS networks, taking into account physical impairment effects.

Because the offset time of bursts varies in OBS networks, the voids or fragmentation on the channels in the outgoing links can severely degrade the network throughput and blocking probability performance, if not dealt with carefully. A signalling architecture called Dual-header Optical Burst Switching (DOBS) is proposed to reduce the scheduling algorithm complexity. We study the burst scheduling problem and propose an impairment aware scheduling algorithm in DOBS networks.

QoS provisioning is an important issue in OBS networks. We have dealt with relative QoS support problem and proposed a QoS provisioning algorithm subject to the physical impairment constraints. A high-priority burst requires a better quality of service in terms of blocking probability, and at the same time, the transmission of the burst should satisfy physical impairment constraints.
# Table of Contents

1 Introduction ........................................... 1
   1.1 WDM Optical Networks ................................ 1
   1.2 Optical Circuit Switching ............................. 4
   1.3 Optical Packet Switching .............................. 8
   1.4 Optical Burst Switching ............................... 13
   1.5 Dissertation Focus .................................. 17
   1.6 Organization of Dissertation .......................... 20

2 Optical Burst-Switched Networks ....................... 21
   2.1 Burst Assembly ........................................ 21
   2.2 OBS Signalling Protocols .............................. 23
      2.2.1 TAW .............................................. 24
      2.2.2 TAG .............................................. 25
      2.2.3 JIT .............................................. 26
      2.2.4 JET .............................................. 27
   2.3 Contention Resolution ................................. 29
      2.3.1 Optical Buffering .................................. 30
      2.3.2 Wavelength Conversion ............................ 31
      2.3.3 Deflection Routing ................................. 32
      2.3.4 Burst Segmentation ................................. 32
      2.3.5 Feedback-Based Contention Resolution .......... 33
   2.4 Burst Scheduling and Grooming Algorithms .......... 33
      2.4.1 Burst Scheduling ................................... 33
      2.4.2 Burst Grooming .................................... 35
   2.5 QoS Provisioning in OBS .............................. 36
   2.6 Summary .............................................. 38

3 Per-Hop Traffic Grooming in Optical Burst-Switched (OBS) Networks .................. 39
   3.1 Introduction .......................................... 39
   3.2 Problem Description ................................... 40
   3.3 Integer Linear Program Formulation ................. 42
      3.3.1 Notation .......................................... 43
      3.3.2 ILP1: Proportional Delay Slack Distribution with No Wavelength Conversion or FDLs .......... 44
      3.3.3 ILP2: Proportional Delay Slack Distribution with Wavelength Conversion Only .......... 45
      3.3.4 ILP3: Proportional Delay Slack Distribution with FDLs Only .................... 47
# List of Figures

1.1.1 WDM network evolution. ................................. 3
1.1.2 Illustration of an MPLS-enabled network. ................................. 4
1.1.3 Different switching technologies in WDM networks. ................................. 5
1.2.4 Process of an OCS flow. ................................ 6
1.2.5 Illustration of an optical wavelength-routed network architecture. ...................... 7
1.3.6 An optical packet-switched network. ................................ 9
1.3.7 Illustration of OPS node architecture. ................................ 9
1.4.8 An example of optical burst switching. ................................ 14
1.4.9 Burst process flow - the offset time should be larger than the sum of control packet processing time. ................................ 15
1.4.10 Illustrations of (a) optical packet switching, and (b) optical burst switching. ........... 16
2.1.1 An optical burst switched network. ................................ 21
2.1.2 Burst assembly. ............................................. 22
2.1.3 Burst disassembly. ............................................. 22
2.2.4 Two approaches to separation of control packet and data burst. ...................... 24
2.2.5 Tell-and-wait reservation. ............................................. 25
2.2.6 Tell-and-go reservation. ............................................. 26
2.2.7 Operation of wavelength with immediate reservation. ............................................. 27
2.2.8 JIT process flow. ............................................. 28
2.2.9 JET process flow. ............................................. 29
2.2.10 Operation of wavelength with delayed reservation. ............................................. 29
3.4.1 Pseudo code for traffic grooming. ............................................. 53
3.4.2 Pseudo code for grooming bursts on the same wavelength. ................................. 54
3.4.3 Pseudo code for grooming a burst with some bursts on a specified wavelength and within a specified delay bound. ................................. 54
3.4.4 Pseudo code for grooming a burst with some bursts on a specified wavelength and without additional delay. ............................................. 55
3.4.5 Pseudo code for deciding whether two large bursts can be groomed. ...................... 55
3.4.6 Pseudo code for finishing grooming first using FDLs followed by using wavelength conversion. ............................................. 56
3.4.7 Pseudo code for grooming two bursts on a third wavelength with void filling. ........... 57
3.5.8 Grooming performance with various packet sizes ($T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$). ............................................. 60
3.5.9 Grooming performance with various switching time (PacketSize = 600, 
nCC = 2, nDC = 3, nFDL = 5, δ = 10µs). ........................................ 61
3.5.10 Grooming performance with various number of FDLs (PacketSize = 600, 
T_x = 10µs, nCC = 2, nDC = 3, δ = 10µs). ................................. 62
3.5.11 Grooming performance under various FDL granularities (PacketSize = 
600, T_x = 10µs, nCC = 2, nDC = 3, nFDL = 5). ......................... 63
3.5.12 Grooming performance with various number of data wavelengths (Packet- 
Size = 600, T_x = 10µs, nCC = 2, nFDL = 5, δ = 10µs). ............... 64
3.5.13 Grooming performance with different combinations of absence/presence 
of FDLs and wavelength conversion (PacketSize = 600, T_x = 10µs, 
nCC = 2, nDC = 3, nFDL = 5, δ = 10µs). .......................... 65

4.2.1 An example of edge node architecture. ................................. 68
4.2.2 An example of core node architecture. ................................. 69
4.2.3 An illustration of burst grooming exploring node light splitting capability. 70
4.3.4 Pseudo code for burst grooming with no over-routing waste. ....... 79
4.3.5 Pseudo code for obtaining a set of sub-bursts that have the largest total 
burst length. ............................................................................ 80
4.3.6 Pseudo code for trimming a set. ........................................... 80
4.3.7 Pseudo code for burst grooming that minimizes the relative total resource 
ratio. ....................................................................................... 85
4.4.8 14-node NSF network topology. ......................................... 86
4.4.9 16-node 30-edge network topology. ..................................... 87
4.4.10 70-node 70-edge network topology. .................................... 88
4.4.1 Burst blocking probability under different grooming algorithms for NSF 
network topology. ................................................................. 89
4.4.12 Burst blocking probability under different grooming algorithms for 16-node 
30-edge network topology. ....................................................... 90
4.4.13 Burst blocking probability under different grooming algorithms for 40-node 
70-edge network topology. ....................................................... 91
4.4.14 Average burst end-to-end delay under different grooming algorithms for 
NSF network topology. ............................................................ 92
4.4.15 Average burst end-to-end delay under different grooming algorithms for 
16-node 30-edge network topology. ........................................... 93
4.4.16 Average burst end-to-end delay under different grooming algorithms for 
40-node 70-edge network topology. ........................................... 94
4.4.17 Average number of sub-bursts per groomed burst under different grooming 
algorithms for NSF network topology. ..................................... 95
4.4.18 Average number of sub-bursts per groomed burst under different grooming 
algorithms for 16-node 30-edge network topology. .................... 96
4.4.19 Average number of sub-bursts per groomed burst under different grooming 
algorithms for 40-node 70-edge network topology. .................... 97
4.4.20 Padding waste under different grooming algorithms for NSF network topol-
yogy. ....................................................................................... 97
4.4.2 Padding waste under different grooming algorithms for 16-node 30-edge network topology. .................................................. 98
4.4.2 Padding waste under different grooming algorithms for 40-node 70-edge network topology. .................................................. 98
4.4.2 Over-routing waste under different grooming algorithms for NSF network topology. .......................................................... 99
4.4.2 Over-routing waste under different grooming algorithms for 16-node 30-edge network topology. .................................................. 99
4.4.2 Over-routing waste under different grooming algorithms for 40-node 70-edge network topology. .................................................. 100

5.3.1 Physical impairment aware burst scheduling process in an OBS network. 110
5.3.2 Flow diagram of the JET based physical impairment constrained algorithm (JETPIC). ...................................................... 111
5.3.3 Flow diagram of the integrated physical impairment constrained algorithm (IPIC). ...................................................... 114
5.3.4 Flow diagram of the enhanced integrated physical impairment constrained algorithm (EIPIC). ........................................ 116
5.4.5 The 14-node NSF network topology. ................................................. 118
5.4.6 The 16-node Torus network topology. ............................................. 119
5.4.7 The 8-node ring network topology. .................................................. 120
5.4.8 Burst blocking probability under different scheduling algorithms for NSF network topology. .................................................. 121
5.4.9 Burst blocking probability under different scheduling algorithms for Torus network topology. .................................................. 122
5.4.10 Burst blocking probability under different scheduling algorithms for ring network topology. .................................................. 123
5.4.11 Average burst end-to-end delay under different scheduling algorithms for NSF network topology. ........................................... 124
5.4.12 Average burst end-to-end delay under different scheduling algorithms for Torus network topology. ........................................... 125
5.4.13 Average burst end-to-end delay under different scheduling algorithms for ring network topology. ........................................... 126

6.1.1 Illustration of the dual-header OBS signalling. ......................... 131
6.1.2 The end-to-end signalling in DOBS networks. ......................... 132
6.3.3 Example of phantom burst resource waste in DOBS networks. .... 138
6.3.4 Components of burst scheduling. .................................................. 139
6.3.5 Example of admission control. ..................................................... 140
6.3.6 Flow diagram of QoT verification. ................................................ 144
6.4.7 Burst blocking probability under different scheduling algorithms for the NSF network topology. ........................................... 148
6.4.8 Burst blocking probability under different scheduling algorithms for Torus network topology. ........................................... 149
6.4.9 Burst blocking probability under different scheduling algorithms for ring network topology. .................................................. 150
6.4.10 Average burst end-to-end delay under different scheduling algorithms for the NSF network topology. ........................................... 151
6.4.11 Average burst end-to-end delay under different scheduling algorithms for Torus network topology. .................................................. 152
6.4.12 Average burst end-to-end delay under different scheduling algorithms for ring network topology. .................................................. 153

7.2.1 An example that shows resources are not available for high priority bursts in DOBS networks. .................................................. 159
7.2.2 Flow diagram of the proposed QoS supporting burst scheduling algorithm. 160
7.3.3 Burst blocking probability under QoS provisioning scheduling algorithms for the NSF network topology in DOBS and single-header systems. . 164
7.3.4 Burst blocking probability under QoS provisioning scheduling algorithms for Torus network topology in DOBS and single-header systems. . 165
7.3.5 Burst blocking probability under QoS provisioning scheduling algorithms for ring network topology in DOBS and single-header systems. . 166
7.3.6 Average burst end-to-end delay under QoS provisioning scheduling algorithms for the NSF network topology in DOBS and single-header systems.167
7.3.7 Average burst end-to-end delay under QoS provisioning scheduling algorithms for Torus network topology in DOBS and single-header systems. 168
7.3.8 Average burst end-to-end delay under QoS provisioning scheduling algorithms for ring network topology in DOBS and single-header systems. 169

xii
List of Tables

3.1 Grooming Performance Comparison between ILP and Heuristic for One Specific Burst Set (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$, $\alpha = 0.1$, $\beta = 0.1$). ........................................... 59

3.2 Average Grooming Performance Comparison between ILP and Heuristic for 100 Burst Sets (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$, $\alpha = 0.1$, $\beta = 0.1$). ........................................... 59
I would like to express my gratitude to my advisor, Dr. Bin Wang. It is not often that one finds an advisor and colleague that always finds the time for listening to the little problems and roadblocks that unavoidably crop up in the course of performing research. His technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general.

My thanks also go to the members of my major committee - Dr. Krishnaprasad Thirunarayan, Dr. Yong Pei, Dr. Yuqing Chen and Dr. Waleed Smari for their precious time in reviewing the manuscript and their valuable suggestions. I am also grateful to them for their encouragement and support on my research and studies.

I am very fortunate to work with a group of students in the Advanced Networking Research Lab at Wright State, including Tianjian Li, Xubin Luo, Ashish Deshmukh, Tao Zhang, and Giovani Rimon Abuaitah. I had a great time working with them.

I have been supported financially by the Dayton Area Graduate Studies Institute (DAGSI) and the Department of Energy Early Career Award during the past three and half years. I would like to express my gratitude to DAGSI and DoE for the
generous support.

Finally, I give my deepest gratitude and love to my parents. My parents have encouraged and assisted me constantly throughout my life. Their faith in me has helped during times when I have doubted my own abilities, and they have helped out in many times of need.
Chapter 1

Introduction

1.1 WDM Optical Networks

The Internet and the Internet Protocol (IP) are currently the basis for most of the network communications. The telecommunications industry has experienced extraordinary changes during the past 20 years, and there is no indication that the exponential data traffic growth will stop, since the fundamentals behind the Internet revolution continue to remain so strong. The applications requiring large bandwidth include Voice-over-IP (VoIP), online video, online gaming, dynamic navigation systems, tele-medicine, e-Science, e-Astronomy, and so on [4]. Furthermore, the number of Internet hosts continues to increase by 30 percent each year, which may result in around a 70% increase in the number of connections [5].

Therefore, the telecommunication networks must be able to provide huge and increasing capacities. Wavelength division multiplexing (WDM) optical networks [1] have been considered as an efficient and cost-effective way to cope with the increasing capacity requirement. It has been demonstrated that Tbit/s level traffic can be provided by the optical links over long distance. For example, an ultra-dense WDM
link can transmit data at the speed of 10.95 Tbit/s [6]. With WDM technology, each fiber consists of multiple virtual fibers, since WDM divides an optical fiber into multiple non-overlapping wavelengths. Usually, a fiber can be composed of several tens of wavelengths. With the technology of Dense Wavelength-Division Multiplexing (DWDM), a fiber can include up to several hundred or a thousand wavelengths [7, 8]. The advantages of WDM optical networks include scalability, reliability, transparency, simplicity, and so on and so forth.

The replacement of point-to-point wavelength division multiplexing (WDM) links for copper cables offers much higher bandwidth and is less susceptible to various kinds of electromagnetic interferences and other undesirable effects [9]. In the high capacity point-to-point optical links, the data is transmitted in the optical domain; however, the data needs to be converted to the electrical domain at the nodes for switching and signal processing. Consequently, there is a great mismatch between the high capacity optical links and electronic switching and processing technologies.

Today, the optical communication is evolving from the simple point-to-point optical links to more intelligent optical networks which can perform more functions in the optical domain, such as routing and wavelength assignment. In the long run, the optical networks will be capable of routing and switching optical packets as shown in Fig. 1.1.1. The optical networks also need to overcome the physical impairments in the optical fibers, such as, dispersion, noise, nonlinear effects, and so on.

Some work has been devoted to enhancing the Internet Protocol (IP) to support traffic engineering [10, 11] and Quality of Service (QoS) [12]. In order to simplify the forward function to support fast switching, QoS support and traffic engineering,
Figure 1.1.1: WDM network evolution.

Label Switching Routers (LSRs) and Multi-Protocol Label Switching (MPLS) [13, 14] are used as illustrated in Fig. 1.1.2. In order that optical cross-connects (OXC)s can provide some switching functions, the Internet Engineering Task Force (IETF) has extended MPLS to the Generalized MPLS (GMPLS) [15] to support the multiple types of routing and management of traffic demands, including the setup and tear-down of lightpaths, which is usually based on wavelength switching and referred to as Multiple-Protocol Lambda Switching (MPλS) [16, 17]. GMPLS offers an integrated control plane based on IP to support optical networks.

However, the nature of the Internet traffic, which includes burstiness of the connection durations, self-similarity, and asymmetry [18], has required an IP-centric network architecture. The range of future services will be very diverse in terms of required bandwidth, channel occupancy, duration set-up time and frequency. The wavelength based switching offers the granularity of switching at a fiber, waveband,
or a wavelength level, which may not satisfy the desired fine granularity and flexibility [19].

Optical Packet Switching (OPS) provides a counterpart of packet switching in optical domain. However, OPS is still deemed technically infeasible and its competitiveness as a backbone technology is debatable. For example, the header cannot be processed all optically at reasonable cost, the header and payload need to be synchronized, complex optical buffers are needed, and so on. Consequently, OBS, a compromise between OCS and OPS shown in Fig. 1.1.3, is an alternative approach to satisfy the required flexibility, efficiency, and high bandwidth utilization [20, 21].

### 1.2 Optical Circuit Switching

Optical circuit switching (OCS) [22, 23] paradigm explores the wavelength functionality for routing by establishing end-to-end lightpaths between node pairs, and pursues a wavelength-routed network architecture with a whole wavelength as its finest
switching granularity. There is a dedicated lightpath for each request (call). Each lightpath may span multiple fiber links to provide a circuit-switched interconnection between the source and destination nodes. In general, circuit switching has three distinct phases: circuit set-up, data transmission and circuit tear-down as shown in Fig. 1.2.4.

One of the main features of circuit switching is its two-way reservation process in the first phase in which a source sends a request for setting up a circuit and then expects an acknowledgment from the corresponding destination. A circuit is set up by reserving a fixed bandwidth channel on each link along a path from the source to its corresponding destination. Another feature of circuit switching is that all the intermediate switches will be configured, so that the channels on the adjacent links can establish and remain a circuit for the duration of the call. Due to this feature, there is no need for buffer at any intermediate node [19].

In WDM networks, circuit switching can be built on the concept of wavelength
routing, where an all-optical wavelength path, consisting of a dedicated wavelength channel on every link, is established between two edges of the network. A wavelength routing network, shown in Fig. 1.2.5, consists of wavelength cross-connects interconnected by a set of fiber links. A request is accommodated in the form of a lightpath traversing a set of fiber links in the optical network. Information transmitted on a lightpath does not need to take optical-electrical-optical (O/E/O) conversion, which means the data transmission is transparent.

As shown in Fig. 1.2.5, the information from the client subnetwork is collected at the edge node and sent to the optical core network via the user-to-network interface (UNI), which is the interface between the client subnetwork and the optical core network. A lightpath starts from one edge node and ends at another edge node; the data may be in the electrical form in the client subnetwork. Edge nodes supporting
Figure 1.2.5: Illustration of an optical wavelength-routed network architecture.

Various network topologies, including ring and mesh networks, can be connected to the core network, on the condition that the UNI is appropriately defined. By using the OXC power splitting capability, the lightpath concept can be generalized to a light-tree, which has one source node and multiple destination nodes [22, 24, 25]. An OXC equipped with power splitting capability is denoted as a multicast-capable OXC.

Optical circuit switching is especially suitable for supporting Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH) communication. The reasons are [26]: (1) SONET/SDH switches communicate with each other at a constant bit rate that matches the bandwidth of a wavelength; (2) the connection duration is long relative to the path set-up time; (3) the number of expensive SONET switches can be reduced with proper traffic grooming and wavelength assignment algorithms; and (4) the optical switches (wavelength routers) based on opto-mechanical, acousto-optic or thermo-optic technologies are currently too slow for efficient packet-switching.

Circuit switching (of wavelength channels) is relatively simple to realize, and it
is reliable. However, OCS requires a long set-up time for channel establishment and release regardless of the connection holding time. This overhead is mainly determined by the end-to-end signalling time, and it leads to poor channel usage if connection holding times are very short. For long holding times, circuit switching is very efficient from a signalling overhead point of view, and it is good for smooth traffic and QoS guarantee due to fixed bandwidth reservation.

However, in IP-centric networks with its bursty behavior [27], OCS provides coarse granularity bandwidth and can not adapt to the dynamic traffic well, which leads to inefficient resource utilization. In this case, OCS either wastes bandwidth during off/low-traffic periods, or needs too much overhead (e.g., delay) due to frequent set-up/release (for every burst). Furthermore, large buffers may be needed at the network ingress to store the incoming packets waiting for the acknowledgement of successful lightpath setup.

1.3 Optical Packet Switching

In contrast to circuit switching which sets up a constant bit rate and constant delay connection between two nodes for their exclusive use for the duration of the communication, packet switching paradigm transmits data in unit of packet, where packets are queued or buffered at intermediate nodes and routed over the network links to get to the destination. Therefore, different packets will experience different delays. Packets can be switched based on datagrams or virtual circuits. In both scenarios, the traffic at the intermediate nodes is processed packet by packet, and the transmission paradigm of packet switching is store-and-forward. Therefore, buffer space is required
Optical packet switching (OPS) [28–34] is the counterpart in optical domain of packet switching, and it provides packet switching in optical domain as illustrated in Fig. 1.3.6. Optical packet switching is suitable for supporting bursty traffic since it allows statistical sharing of the channel bandwidth among packets belonging to different source and destination pairs. Several projects have been carried out in OPS [35–38].

A WDM optical packet switch consists of four parts: the input interface, the
switching fabric with associated optical buffers and wavelength converters, the output interface, and the control unit as shown in Fig. 1.3.7. The input interface focuses on packet delineation and alignment, packet header information extraction and packet header removal. The switching fabric is the core of the switch; it routes the packets to their proper outputs and executes contention resolution. The output interface inserts a new header and may have to regenerate the optical signal. The switching is controlled by the control unit with the information in the packet header. Normally, the packet size is fixed because of the synchronization requirements [39].

Packets arriving on an input fiber are first demultiplexed into individual wavelengths and then sent to the input interface for processing. Each packet consists of two parts: payload and header. The header and the payload of the packet are separated. The header is processed by the control unit electronically or optically (though the optical logic is very primitive), while the payload remains in the optical domain. Ideally, the packet would be processed all in the optical domain. However in practice, some functions, such as processing the header and controlling the switch, are processed in the electrical domain. This is because very limited processing capabilities in the optical domain are available. The switch control unit determines an appropriate output port and wavelength for the packet, and instructs the switch fabric to route the packet accordingly. During the routing of the packet, optical buffer and/or wavelength converters may be needed. After being processed, the new packet header updated by the control unit will be combined with the payload again at the output interface. The new optical packet will be forwarded to the next node on its outgoing link. In OPS networks, data is transmitted in the form of optical packets
[40], and an optical packet is the finest granularity that OPS provides. The packets are transmitted in the optical core network without being converted to electrical domain by using optical switches at the intermediate nodes. Packet transmission does not need connection setup, and hence OPS supports bursty traffic better than OCS. The packet transmission allows a good statistical multiplexing, which leads to high network resource utilization if the traffic is bursty.

OPS provides the vision of a bandwidth-efficient, flexible, data-centric all-optical Internet. However, to realize this vision, there are significant challenges since OPS requires practical, cost-effective, and scalable implementations of optical buffering and packet-level parsing [39].

In traditional electronic packet switches, a very simple solution to overcome the contention problem is to buffer contending packets, thus exploiting the time domain. Random access memory (RAM) is available in the electrical domain, so that packets can be stored in the switch until the switch is ready to forward the packets. One of the biggest challenges is that there is no optical equivalence of the electrical random access memory (RAM), and accordingly, an optical data signal can only be delayed for a limited amount of time via the use of fiber-optic delay lines (FDLs). FDLs are just long pieces of fiber to simply delay the packets. A packet cannot be stored indefinitely with FDL, because FDL works in a discrete way in some sense. Furthermore, FDLs will incur additional signal quality degradation.

In order to achieve rapid reconfiguration on a packet-by-packet basis, the switch fabric at an OPS node must be capable of performing the switching on the order of a few nanoseconds at data rates of 40Gbps and beyond. Some optical switch
fabric technologies have been investigated, including those based on opto-mechanical, thermo-optic, or acousto-optic methods [41]. However, most of the optical switch fabric technologies are limited to switching speeds in the millisecond or microsecond range. Two promising technologies, namely semiconductor optical amplifier (SOA) switches and electro-optic lithium niobate (LiNbO3) switches, are capable of switching on the order of a few nanoseconds. Nevertheless, these two technologies can not satisfy the requirements for reliability, cost-effectiveness, low cross-talk radiation, and so on.

Optical networks can be divided into two categories: synchronous (slotted) and asynchronous (unslotted) [42]. In a synchronous optical network, all the optical packets have the same size; while in an asynchronous network, the packets will have various sizes. In asynchronous network [43–45], the arrival packets can be switched at any time without being aligned. In slotted network [46, 47], the duration of a slot is equal to the sum of the packet size and the optical header length and the appropriate guard bands [33]. The disadvantages of synchronous OPS are the needs for optical synchronizer at the switch interfaces and a global reference clock. Packets arriving at a node may not be aligned with local clock due to the variable link propagation delay. Therefore, the input interface needs to synchronize the packets, so that the packets can be aligned to the switching time slots. However, it is not easy to maintain synchronization in the optical domain, because the datagrams need to be segmented into fixed length at the source node and resembled at the destination at a very high speed. Asynchronous OPS switches do not need optical synchronizer to synchronize the incoming packets. Therefore, the hardware is simpler, and the edge node assembly is also simpler [43]. Asynchronous OPS is also more flexible and robust than
synchronous OPS. However, the contention and packet loss will increase, and the network resource utilization will decrease due to the asynchronous operation. Most of the studied cases assume the synchronous OPS network model.

Due to the technical constraints, such as no RAM buffer, the need to synchronize packet header and payload, costly all-optical packet processing, and so on, the OPS is still far from its realization. Therefore, an alternative technology, OBS, which is a hybrid approach of OPS and OCS is proposed to overcome the disadvantages of OCS and OPS to realize a flexible and bandwidth-efficient optical network.

1.4 Optical Burst Switching

The concept of burst switching has been proposed in 1980’s [48, 49]. However, this technology was not successful in the electrical networks due to the requirements and complexity compared to the packet switching techniques. In optical networks, there is a large discrepancy between the optical transmission capacity and electrical processing capability, and there is no random access memory in optical domain available. Therefore, it is beneficial to keep the data in optical domain. Optical burst switching (OBS) was proposed in the late 1990’s [20, 21], and it is a promising technology to combine the advantages of both OCS and OPS, and to avoid their disadvantages at the same time.

OBS is an adaption of an ITU-U standard for burst switching for ATM networks, i.e., ATM block transfer [50]. Optical burst switching is based on the separation of the control plane and the data plane. In OBS networks, a control packet is sent ahead of the data burst with an offset time. The data burst is kept in the optical domain at
the intermediate nodes, while its associated control packet or header is converted into electrical domain for processing. An example of burst transmission in OBS networks is illustrated in Fig. 1.4.8.

Figure 1.4.8: An example of optical burst switching.

Edge node A needs to transmit bursts to edge node B. A control packet is sent from node A to B via the control channel to configure the switches on the data burst transmission path [51]. After an offset time indicated in Fig. 1.4.8, the burst begins to transmit. Note that, the offset time for a burst may be different at different nodes; the offset time for different bursts may also be different. The control packet is converted to electronics for processing at the core nodes to configure the switch, so that the data burst can stay in the optical domain without O/E/O conversion. After processing, the packet will be converted back into optical domain, and then forwarded to the next node, until the control packet gets to the destination B. If an OBS node does not have enough resource to accommodate the burst, the burst will be dropped. The offset time should be large enough, so that the control packet can be processed and
the switch can be configured before the data bursts arrive as shown in Fig. 1.4.9.

![Diagram of burst process flow]

Figure 1.4.9: Burst process flow - the offset time should be larger than the sum of control packet processing time.

This out-of-band signalling scheme is opposed to that in OPS as shown in Fig. 1.4.10. Each data burst has an associated control packet containing the header information, such as burst length, burst arrival time, burst incoming wavelength index, and so on. Control packet information is transmitted on dedicated control channels. Only a very small number of control channels are needed, because control packets are significantly smaller than data bursts in general.

At the source edge nodes, data packets are aggregated into a large packet, *burst*, for transmission, which will be further discussed later. The burst will be disassembled to original data packets at the destination edge node. Each burst may contain several packets, such as IP packets, ATM cells, and so on. This aggregation of packets allows amortization of the switching overhead across multiple packets. Therefore, OBS can be viewed as lying between OCS and OPS. During burst assembly and disassembly, the packets will be buffered in the electrical RAM.
Several OBS signalling protocols have been proposed, such as tell-and-go (TAG), tell-and-wait (TAW), just-enough-time (JET) [20], and just-in-time (JIT) [51, 52]. Some of the common characteristics are as follows [19]:

- Client data (IP packets) are aggregated and disassembled at the edge nodes, and the statistical multiplexing at the burst level is achieved in the core of the OBS network.

- Control information (header) and user information (data) are separated in space and time, and costly O/E/O conversion is only required in a few control chan-
1.5 Dissertation Focus

OBS provides dynamic bandwidth allocation and statistical multiplexing, and OBS is more feasible to implement than OPS, since it has fewer technical restrictions. OBS reduces the total overhead by aggregating the client packets at the edge nodes. When client packets with similar characteristics arrive at the edge node, such as destination, quality-of-service, etc., they are assembled into a burst.

In this dissertation, we will explore several important and challenging issues in OBS networks in order to improve the utilization of network resource:

We have studied two traffic grooming problems in OBS networks: (1) per-hop traffic grooming, and (2) burst grooming by exploring node light-splitting capability.

To reduce the switching overhead, small bursts may be groomed to reduce resource waste and switching penalty. We have studied the per-hop burst grooming problem where bursts with the same next hop may be groomed together, assuming all the network nodes have the grooming capability. Our objective is to minimize the number of formed larger bursts, and at the same time, to reduce the number of used wavelength converters (if wavelength conversion is available) and the number of used FDLs (if FDLs are available), that is, to strike a proper balance between burst grooming and grooming cost.

In order to reduce computation overhead and processing delay incurred at the core nodes, we assume that grooming can only be performed at edge nodes and the core node can send a burst to multiple downstream links, that is, the core node has
light-splitting capability. We have attempted to groom small bursts into larger bursts, and select a proper route for each large burst, such that total network resources used and/or wasted for delivering the small bursts is minimized.

In all-optical networks, an optical signal may traverse a number of intermediate nodes and long fiber segments without any optical-electrical-optical (O/E/O) conversion, which significantly reduces the overall network cost. Most of the network designers of all optical networks assume an ideal physical optical network, that is, the signal quality is immune to noise and signal quality degradation. However, optical signal transmission quality is subject to various types of physical impairment introduced by optical fibers, switching equipment, or other network components. The signal degradation due to physical impairment may be significant enough such that the bit-error rate of received signals is unacceptably high at the destination, rendering the signal not usable. A burst will be dropped when the quality of the signal has fallen below the requirement. Based on earlier work, we have studied the burst scheduling problem and proposed three effective burst scheduling algorithms in OBS networks, taking into account physical impairment effects. At an OBS node, the proposed algorithms schedule bursts for transmission by searching for available resources as well as verifying signal quality.

In OBS networks, the resource reservation is out-of-band, that is, a single control packet is sent on a control channel ahead of the data burst with an offset time to reserve the network resource. Because the offset time of bursts varies, the burst scheduling may be out of order, which means that the bursts may not be scheduled according to the burst arrival order. Hence there might be some voids or fragmen-
tation on the channels in the outgoing link. These voids can severely degrade the network throughput and blocking probability performance, if not dealt with carefully. In general, a scheduling algorithm with signal quality consideration is more complicated than a corresponding scheduling algorithm counterpart which does not consider physical impairments. A signalling architecture called Dual-header Optical Burst Switching (DOBS) is proposed to reduce the scheduling algorithm complexity. We study the burst scheduling problem and propose an impairment aware scheduling algorithm in DOBS networks. At an OBS node, the proposed algorithm schedules bursts for transmission by searching for available resources using admission control as well as verifying signal quality.

QoS support is an important issue in OBS networks. However, schemes developed for electronic networks can not be applied directly to OBS networks because of the following two reasons. The first reason is that the electronic buffer of bursts needs O/E/O conversion, which is costly and loses the data transparency. The second reason is that no random access memory in optical networks is available. We have dealt with relative QoS support problem subject to the physical impairment constraints. A high-priority burst requires a better quality of service in terms of blocking probability, and at the same time, the transmission of the burst should satisfy its requirement for physical impairment constraints. A low priority burst may be preempted to make resource available for a high priority contending burst in our proposed algorithm.
1.6 Organization of Dissertation

The rest of the dissertation is organized as follows. In this chapter, we introduce WDM networks and describe the basic optical switching technologies. Chapter 2 examines the current research on OBS networks, including burst assembly, signalling protocols, contention resolution, and QoS support. In Chapter 3 we present per-hop burst grooming. Chapter 4 discusses burst grooming by exploring node light-splitting capability. Chapter 5 addresses physical impairment aware scheduling problem in optical burst switched networks. We describe physical impairment aware ordered scheduling algorithm in dual control packets optical burst switched networks in Chapter 6. Chapter 7 tackles QoS provisioning in OBS networks, taking physical impairment into consideration. Finally, Chapter 8 concludes this dissertation with future work.
Chapter 2

Optical Burst-Switched Networks

2.1 Burst Assembly

An OBS mesh network architecture is depicted in Fig. 2.1.1.

The finest data transmission unit is a burst. Before transmission, client data are aggregated into bursts at the edge nodes as shown in Fig. 2.1.2. Therefore, the edge nodes are the interface between a client network and an OBS core network. Burst assembly is the procedure of aggregating packets from various sources, such
as IP packets, into bursts at the edge of an OBS network. The reverse procedure is called burst disassembly as illustrated in Fig. 2.1.3. Both burst assembly and burst disassembly are performed in the electrical domain.

![Burst assembly diagram](image1)

**Figure 2.1.2: Burst assembly.**

![Burst disassembly diagram](image2)

**Figure 2.1.3: Burst disassembly.**

Various burst assembly schemes have been proposed [53–55]. Different assembly schemes may have different impacts on the burst length distributions, number of client packets per burst and inter-arrival time between bursts. The impact of burst assembly schemes on the burst size and burst arrival rate is studied in [56–58]. The use of burst assembly to support Quality of Service (QoS) is also studied [59–61].

There are mainly three ways to conduct burst assembly: (1) a timer-based approach [54]; (2) a length-based approach [62]; and (3) a hybrid scheme [55, 56, 63, 64].

In timer-based burst assembly approaches, a burst is created and sent into the optical network when a timer expires so that the burst assembly works in a periodic
way. Consequently, the network may have variable length bursts, and the length of
the bursts varies with the load changes.

In length-based approaches, a burst is generated upon the arrival of enough bytes
of packets, which results in the same or very similar length bursts. That is, the burst
length distribution will have a very low variance.

In hybrid approaches, the timer-based approach and the length-based approach
are combined together, that is, a burst is formed either when a timer expires or a
predetermined length threshold has been reached.

2.2 OBS Signalling Protocols

In OBS, signalling is based on the separation of control packet and data burst (pay-
load). The control packet is sent first to inform the core nodes and configure the
switches, while the transmission of data bursts is delayed by an offset time. This
offset time is to compensate for some delays in the OBS core, such as control packet
processing delay and propagation delay, etc. In this way, the control packet can
be processed in electronics, while the data burst can remain in the optical domain.
Sometimes, FDLs may be used to delay the bursts, in order to ensure that the control
packet is processed before the data burst arrival as shown in Fig. 2.2.4.

In OBS core, one-way reservation is used in most common approaches to burst
transmission path set-up, that is, the data burst will be transmitted without waiting
for the acknowledgement [65]. Various reservation schemes have been proposed in
OBS, such as tell-and-wait (TAW) [66, 67], tell-and-go (TAG) [68, 69], Horizon [21],
just-in-time (JIT) [51, 52], just-enough-time (JET) [20], and so on.
2.2.1 TAW

TAW comes from one of the variants of the ATM block transfer scheme (ABT), and it is basically a two-way reservation scheme. Before data burst transmission, a control packet is sent from the source to the destination to reserve the required resource (bandwidth) at each node along the transmission path. At each node, the requested resource is reserved if available. If the requested resource is granted along the path, then the destination will send an acknowledgement (ACK) to the destination in the reverse direction of the burst transmission path. Otherwise, a negative acknowledgement (NACK) will be sent to the source. On receiving the NACK, the reserved upstream resource for this burst will be released.

Upon the receipt of the ACK, the source node transmits the burst into the OBS core network. As a result, the burst will go through the path without worrying about dropping. The resource reserved for the burst at a core node will be released when the burst transmission is completed at the node. The offset time in this case can be
seen as the round trip time of the control packets. When an NACK is received, the source will try to make another request to send the burst at a later time. This scheme is depicted in Fig. 2.2.5.

![Figure 2.2.5: Tell-and-wait reservation.](image)

### 2.2.2 TAG

TAG is inspired by another variant of the ABT, i.e., fast reservation protocol (FRP) in ATM or ATM block transfer with Immediate Transmission (ABT-IT) [20]. In this scheme, the source node sends the bursts without making any resource reservation in advance. A copy of the burst will be kept at the source node until the source has received an ACK from the destination.

At an intermediate node, the burst needs to be delayed for control packet processing and switch configuration. Therefore, buffer space may be needed to hold the incoming bursts at the intermediate nodes. If the reservation at an intermediate node fails, an NACK will be sent back to the source to initiate the retransmission of
the burst, and then the source will initiate the retransmission at some time later. If a burst has successfully reached the destination, the destination will send an ACK back to the source to confirm the successful transmission. This scheme is depicted in Fig. 2.2.6.

![Figure 2.2.6: Tell-and-go reservation.](image)

### 2.2.3 JIT

JIT is a one-way reservation scheme. Just-in-time means that the switching fabric has already been configured by the time a burst arrives. JIT adopts an immediate approach to reserve resource, that is, the resource is immediately reserved after the control packet is processed. The reserved channel will remain allocated for the burst until the burst finishes its transmission. If the required channel can not be reserved at that time, the reservation fails.

This immediate reservation is illustrated in Fig. 2.2.7. At a specific time, the channel can have one of the two states: (1) reserved, and (2) free. When burst
i arrives at time $t_1$, it is successfully scheduled, and the resource is immediately reserved. That is, the resource will remain reserved for burst $i$ from time $t_1$ to $t_3$, although the arrival time of burst $i$ is $t_2 > t_1$. After time $t_3$, the channel is available, until another burst $i+1$ arrives. The channel will be reserved for burst $i+1$ from time $t_4$ to $t_6$. If a control packet comes after time $t_1$ and requests the resource between $t_1$ and $t_2$, the request will fail, since the channel has been reserved for burst $i$.

![Figure 2.2.7: Operation of wavelength with immediate reservation.](image)

The JIT reservation protocol is shown in Fig. 2.2.8. The resource reservation works in an explicit-setup-and-explicit-release way, that is, the switch is immediately configured by a SETUP message, and released by an explicit RELEASE message.

### 2.2.4 JET

JET is the most prevailing distributed protocol for OBS networks today. JET is also a one-way reservation protocol. Unlike JIT, JET adopts delayed reservation, that is, JET uses estimated configuration with the information embedded in the control packet. The reservation starts at the expected arrival time of the burst, instead of the control packet’s arrival time.

The rationale of JET is illustrated in Fig. 2.2.9. The control packet is sent to
the destination ahead of the burst transmission. The control packet is processed at each intermediate node to reserve the resource. The switch is configured for the burst according to the information embedded in the control packet, such as the offset time. The release is in an estimative way, which is different from JIT. The switch releases the resource according to the estimated burst departure time.

Delayed reservation is shown in Fig. 2.2.10. The channel is reserved for the first burst starting from the burst arrival time to the burst departure time. In Case 2, both JIT and JET can accommodate the second burst, because the resource has been released by the first burst. In Case 1, JIT can not accommodate the second burst, because the resource is reserved for the first burst during its offset time, although the channel is not used for transmission during that period of time. However, JET can serve the second burst due to the delayed reservation, since resource is only reserved for the transmission time of the first burst.
JET can achieve better blocking performance. However, the switch hardware becomes significantly more complex [70]. JIT is more amenable to hardware implementation [71].

2.3 Contention Resolution

As we have discussed, bursts are sent without an acknowledgement of successful path set-up (one-pass reservation) in most OBS approaches. Thus burst loss can occur in case of contention. Contention occurs at a switch whenever two or more bursts try to
use the same wavelength on the same output link at the same time. When contention occurs, the data carried in the blocked burst are discarded at the node. Therefore, efficient contention resolution in OBS core nodes is essential in order to achieve a low burst blocking probability.

Many contention resolution schemes for OBS have been proposed. These schemes explore the time and/or space domain ability to accommodate bursts to reduce the blocking probability, such as optical buffering [72, 73], deflection routing [74–77], wavelength conversion [78, 79], burst segmentation [80–82], feedback-based contention resolution [83–88], and so on. These schemes above can also be combined together to improve the blocking performance.

2.3.1 Optical Buffering

This approach tries to delay the burst for some time so that a wavelength may be available for the burst. In electronic networks, data can be buffered, and packets involved in contention can be sent at a later time when the output port is available. This process is naturally integrated in the store-and-forward technique.

However, in optical networks, no optical RAM is available. The delay of data is performed through FDLs in OBS [46] to explore the ability in time domain to solve the contention. Approaches have been investigated to design large buffers without a large number of FDLs. Buffer size is increased through cascading multiple stages of FDLs in [89], and through non-degenerate buffers in [90].

In general, the delay ability of FDLs is limited due to the signal quality concerns and physical space consideration as well. It requires more than one kilometer of fiber
to delay a single packet for 5\(\mu s\). Consequently, optical buffers may not be able to cope with high traffic loads.

### 2.3.2 Wavelength Conversion

With wavelength conversion, the bursts involved in contention may be scheduled on a different wavelength in the outgoing link. This approach tries to explore the space domain to solve the contention.

In optical links, there exist several wavelengths. Wavelength conversion converts the incoming wavelength of a burst to another outgoing wavelength. Through wavelength conversion, one or more bursts may be transmitted through different wavelengths so that the bursts will not use the same wavelength on the same link at the same time.

Optical switches may be equipped with different wavelength conversion capabilities. Some categories of wavelength conversion are as follows:

- No conversion: the incoming wavelength can not be converted to another wavelength, that is, the optical data transmission is subject to wavelength-continuity constraint.

- Full conversion: the incoming wavelength can be converted to any outgoing wavelength.

- Limited conversion: wavelength conversion ability is limited, that is, a specific incoming wavelength may be able converted to a limited number of wavelengths.
2.3.3 Deflection Routing

Deflection routing is another approach to reduce the burst blocking probability using the space domain. Instead of scheduling a burst onto a different wavelength in the same outgoing link, deflection routing attempts to schedule the burst with a different route to the destination. That is, the burst is scheduled on an alternative link. The alternative outgoing link forms the first link of a deflection path.

Considering the limited ability of FDLs to delay the bursts, deflection routing may be a good choice. However, the deflected burst may go through a longer path than its preferred path so that the end-to-end delay may be increased.

2.3.4 Burst Segmentation

Normally, the burst that fails in the scheduling is dropped entirely. The rationale of burst segmentation is to break the burst into several segments, and each segment of the burst may include multiple packets. When contention occurs, each segment in the burst may be deflected, dropped, or transmitted on the preferred path.

There are two ways to resolve contention using burst segmentation when contention takes place between two bursts. One is to drop the tail of the first burst, while the other is to discard the head of the second burst. Burst segmentation can decrease the number of bytes dropped. However, it is more complex than other contention resolution methods. It incurs more overhead, because each segment needs its own header information. Multiple segments are sent as a single burst. The network nodes have to recognize the boundaries of each segment. As a result, the switch hardware is more complicated.
2.3.5 Feedback-Based Contention Resolution

The above contention resolution approaches are reactive, that is, the core nodes will try to resolve contention when contention happens. In these approaches, bursts are transmitted with one-way reservation, and there is no feedback on whether the burst transmission is successful or not. Proactive approaches have been proposed to control or avoid contention based on the feedback information.

In these feedback-based schemes, contention is reduced or avoided by dynamically adjusting the data flow at the source to reduce the amount of traffic in the network. Traffic congestion is reduced in [83] by balancing the data traffic between predefined alternative paths. Some traffic can be rerouted to the under-utilized paths to reduce the amount of traffic on the congested links in [84]. In [85], a global load-balancing contention resolution scheme is proposed and the performance is examined for both dynamic and static traffic. TCP-like congestion avoidance mechanism is also applied in OBS networks to regulate the traffic [86–88]. The problem with feedback-based contention resolution is that the overhead is increased.

2.4 Burst Scheduling and Grooming Algorithms

2.4.1 Burst Scheduling

In OBS networks, a key problem is to schedule bursts on output channels. Upon the arrival of a burst, a network core node needs to schedule the burst. The scheduling algorithms proposed can be classified into two classes: without void filling [21, 91] and with void filling. LAUC (latest available unscheduled channel) is a representative of scheduling algorithms without void filling. Scheduling algorithms with void
filling include LAUC-VF (latest available unused channel with void filling) [53], Min-SV (minimum starting void), Min-EV (minimum ending void), Max-SV (maximum starting void), Max-EV (maximum ending void), and so on [92].

In LAUC algorithm, only one real value, the unscheduled time (future available time), is maintained for each data channel. The rationale of LAUC algorithm is to minimize gaps/voids by selecting the latest available unscheduled data channel for each arriving data burst. When a burst with length $L$ arrives at time $t$, the scheduler first finds the outgoing data channels that have not yet been scheduled at time $t$. If there are multiple such channels, the scheduler selects the latest available channel to carry the arriving data burst. The latest available channel is the data channel which has the smallest gap between time $t$ and the end of last data burst just before $t$. The unscheduled time of the selected channel is then updated to $L + t$.

LAUC-VF algorithm is similar to the LAUC algorithm except that the voids can be filled by new arriving data bursts. This algorithm tries to find an unused channel, instead of an unscheduled channel. When a burst with length $L$ arrives at time $t$, the scheduler first finds the outgoing data channels that are available during the time period $(t, t + L)$. If there are multiple such channels, the scheduler selects the latest available channel to carry the arriving data burst. The latest available channel is also the data channel that has the smallest gap between time $t$ and the end of the last data burst before $t$.

Min-SV tries to find an unused channel and explores voids to accommodate bursts. This algorithm searches for a channel that is available for the required transmission period of the incoming burst. Min-SV finds a void which minimizes the difference
between the burst arrival time and the void starting time. This algorithm can be easily adapted to Min-EV, Max-SV and Max-EV.

In general, scheduling algorithms without void filling are simple and have pretty good performance in terms of execution time. However, the algorithms do not keep track of the voids. Hence they are not as good as the scheduling algorithms with void filling in terms of the blocking probability performance. Clearly, the algorithms with void filling have higher time complexity. Therefore, the scheduling algorithm needs to strike a balance between time efficiency and blocking probability performance.

2.4.2 Burst Grooming

To reduce the switching overhead, a minimum burst length requirement, $L_{\text{min}}$, is often imposed in the OBS network where transmitted bursts have to be at least $L_{\text{min}}$ bytes. When the size of a burst is too small to satisfy this minimum burst length requirement, the burst has to be padded, which incurs resource waste, thus potentially increasing burst blocking probability. Therefore, traffic grooming is an important issue when the data burst is small and is comparable to the switching time [93].

We refer to bursts before grooming [94–96] as sub-bursts regardless of their length. Accordingly, we refer to bursts after grooming as groomed bursts. To increase resource utilization in OBS networks, burst grooming can be applied where numerous data bursts are coalesced to form a larger burst that will be switched as one unit in order to reduce the resource waste and switching penalty [93]. For instance, sub-bursts with the same destination can be aggregated into a single burst to reduce the per sub-burst
switching overhead and to use the least number of total switching operations possible for delivering all the sub-bursts. In addition, burst grooming may also reduce inter-burst gaps and recover some channel void capacity, leading to improved network utilization [97].

Different burst grooming schemes have been proposed. Burst grooming can be achieved at the ingress node if sub-bursts are available and destined to the same egress node [93, 97]. Sub-bursts with different destinations can also be groomed at the ingress node and transmitted as a single burst until they are separated at some egress node [98–100].

2.5 QoS Provisioning in OBS

Much work has been devoted to the QoS support in the Internet. Existing schemes are based on packet switching, which adopts the random access memory to store the packets for an arbitrary period of time and provides service differentiation. QoS support is also an important issue in OBS networks. However, schemes developed for electronic networks can not be applied directly to OBS networks because of the following two reasons. The first reason is that the electronic buffer of bursts needs costly O/E/O conversion, resulting in the loss of data transparency. The second reason is that no random access memory in optical networks is available. The bursts are delayed via FDLs which can only delay the bursts for integer units of the FDL granularity. Hence, not all the continuous delays can be implemented.

Two types of approaches have been proposed to provide QoS support [101]. One is relative QoS, where each QoS class is defined relatively in comparison to other
classes, and a higher priority class should experience a lower blocking probability. The other is *absolute QoS*, in which a performance bound, e.g. blocking probability, is provided. An upper bound is guaranteed for the high priority class. The relative QoS model does not provide such a worst-case service level guarantee.

Several schemes have been proposed to support relative QoS in OBS networks. An offset-time-based QoS scheme is proposed in [102, 103] to assign extra offset time to the higher priority classes, without need of FDLs. The QoS scheme implemented in [104] isolates classes of traffic by assigning FDLs that function as optical buffer space, based on the Random Early Detection (RED) technique. An intentional dropping scheme is proposed in [105] in order to give a proportional burst loss probability for different service classes. A preemptive wavelength reservation mechanism is implemented in [106, 107] to provide different degrees of resource assurance to different classes of traffic in proportion to their service classes. QoS is provided in [80] by introducing prioritized contention resolution policies in the network core and a composite burst-assembly technique at the network edge, which resolves contention through prioritized burst segmentation and prioritized deflection, and the burst segmentation scheme allows high-priority bursts to preempt low-priority bursts. A generalized LAUC-VF algorithm is proposed in [108] to improve the QoS performance by prioritizing data bursts, maintaining multiple queues and utilizing limited optical buffer space. A differentiated burst scheduling designed in [109] can adjust the data burst loss rates for different classes of bursts and satisfy differentiated QoS requirement with the available resources by dynamically choosing the early differentiation time and scheduling the high-priority bursts earlier.
There is also some work done for absolute QoS support. An early dropping mechanism is proposed in [110] to probabilistically drop the non-guaranteed traffic. In [101, 110], two mechanisms, early dropping and wavelength grouping, are integrated to enforce a loss probability threshold for guaranteed traffic while reducing the loss rate of non-guaranteed traffic, where the wavelength grouping mechanism provisions necessary wavelengths for the guaranteed traffic. A dynamic virtual lambda partitioning mechanism implemented in [111] supports absolute differentiated services through sharing wavelength resources among several priority lambda groups. The wavelengths are dynamically partitioned depending upon QoS requirements, and the wavelength reservation policies of each priority class are different. For example, high priority traffic can access the wavelength resources within their own priority lambda group as well as the resources in the lower priority lambda groups. Absolute QoS is supported in [112] with the aid of fiber delay lines (FDLs) and a token algorithm. A path clustering technique proposed in [113] groups and prioritizes traffic based on hop-distances between source and destination pairs to provide absolute end-to-end loss probability of guaranteed traffic over the entire network.

### 2.6 Summary

In this chapter, we examine the current development in research on OBS networks, and discuss some important issues in OBS networks. We provide a brief summary of the burst assembly and various OBS signalling protocols. We also discuss scheduling, contention resolution, scheduling, and burst grooming in OBS networks. Furthermore, we introduce some development in supporting QoS in OBS networks.
Chapter 3

Per-Hop Traffic Grooming in Optical Burst-Switched (OBS) Networks

3.1 Introduction

When the arrival rate of data packets at the ingress OBS node is low, a timer-based burst assembly may generate small bursts whose lengths are less than the required minimum burst length. On the other hand, packets may have an end-to-end delivery delay requirement (e.g., IP packets have a soft delivery deadline due to TCP timer’s time-out value). Therefore, a data burst may have an end-to-end delay bound $D$ determined by the packets carried in the data burst, e.g., the least tolerable end-to-end delay of the IP packets in the data burst. In this case, a length-based burst assembly approach may be prevented from forming data bursts of a desired length, rendering it ineffective. As we have discussed in Section 2.4, the overhead is big when the data burst is small comparable to the switching time.

In this chapter, we study the per-hop burst grooming problem where bursts with the same next hop may be groomed together, assuming all the network nodes have
the grooming capability.

The rest of this chapter is organized as follows. Section 3.2 describes the per-hop traffic grooming problem. Integer linear programming formulations of the problem are presented in Section 3.3, the results of which are used for performance comparison against the heuristic algorithms proposed in Section 3.4. Section 3.5 reports the performance evaluation. This chapter concludes in Section 3.6.

3.2 Problem Description

The traffic grooming problem is to coalesce several bursts close in time together to form a larger burst that will be switched as one unit. Previous work [93, 97] studied grooming bursts with the same source and destination nodes. To the best of our knowledge, burst grooming on a per-hop basis, i.e., grooming bursts with the same next hop, has not been looked at. Specifically, an OBS grooming node can combine several bursts with the same next hop to form a larger burst and switch the bursts together. When a larger burst arrives at a node, the node can drop smaller component bursts destined to this node and groom the remaining component bursts with other bursts heading for the same next hop. That is, at a node, some portion of a large burst can be dropped locally, and the rest can be transmitted (or may be groomed with other bursts prior to being transmitted) to the next hop. Furthermore, each node in the network can be equipped with wavelength converters or fiber delay lines (FDLs) which can be utilized to aid burst grooming.

We consider an end-to-end delay bound $D$ to be guaranteed for a burst. If the path dependent end-to-end propagation delay is $d_p$, $D \geq d_p$. Therefore, the delay
slack that may be used by burst grooming, e.g., FDLs, is $d_s = D - d_p$. Assume that a burst goes through an $m$-link path. There are two ways to distribute the delay slack $d_s$: (1) **Proportional distribution** where $d_s$ is divided by $m$ proportionally, that is, a burst can be delayed up to $\frac{d_s}{m}$ at each node that the burst goes through. (2) **Collective distribution** which allows a burst to be delayed an arbitrary amount of time as long as the sum of all delays at each node does not exceed $d_s$. By appropriately setting $D$ (i.e. setting $d_s$), we can meet different burst delay requirements. For example, for synchronous data bursts, we can set $d_s$ to 0 or a small amount of time; for asynchronous data bursts, we can set $d_s$ to a larger value. In this work, we assume that the switching time $T_x$ at each node is the same.

In the following, we consider eight burst grooming cases depending on how per-hop delay slack is distributed and whether or not FDLs or wavelength converters are used. Specifically, in cases 1-4, $d_s$ is proportionally distributed while $d_s$ is collectively distributed:

1. $d_s$ is distributed proportionally and no wavelength converter or FDL is used.
   
   In this case, since a node is not equipped with wavelength converters or FDLs, a burst can not be delayed or converted to use another wavelength;

2. $d_s$ is distributed proportionally and burst grooming can use wavelength conversion only. In this case, a node is equipped with only wavelength converters, can convert the wavelengths of the bursts, but can not delay the bursts;

3. $d_s$ is distributed proportionally and burst grooming can use FDLs only. In this case, a node is equipped with FDLs, can delay a burst up to $\frac{d_s}{m}$ units, but can
not convert the wavelengths of the bursts;

4. $d_s$ is distributed proportionally and burst grooming can use wavelength converters and FDLs. In this case, a node is equipped with both wavelength converters and FDLs, can delay a burst up to $\frac{d_s}{m}$ units, at the same time, it can convert the wavelengths of the bursts.

5. $d_s$ is collectively distributed and no wavelength converter or FDL is used. This case is the same as case 1;

6. $d_s$ is collectively distributed and burst grooming can wavelength converters only. This case is the same as case 2;

7. $d_s$ is collectively distributed and burst grooming can use FDLs only. In this case, a node is equipped with FDLs, can delay a burst up to $d_s$ units, but it can not convert the wavelengths of the bursts. The delay slack $d_s$ of a burst will be updated at each node the burst goes through;

8. $d_s$ is collectively distributed and burst grooming can use wavelength converters and FDLs. In this case, a node is equipped with both wavelength converters and FDLs, can delay a burst up to $d_s$ units; at the same time, it can convert the wavelengths of the bursts.

3.3 Integer Linear Program Formulation

Burst grooming in OBS networks is formulated into 4 integer linear programming (ILP) problems taking into consideration the presence/absence of FDLs and wave-
length conversion in [93]. However, the formulations do not consider the cost of FDLs and wavelength conversion. In their formulations, while the control packets on all the wavelengths are considered simultaneously, the conflicts among bursts are not considered. For example, the formulations cannot deal with the following 2 cases: (1) bursts $n$ and $n + 1$ are overlapping in time; (2) bursts $n$ and $n + 1$ are on different wavelengths and their inter-burst gap is small, but they cannot be groomed together because it will incur conflict to change either wavelength of the two bursts. Furthermore, the delay of a burst has no bound, that is, the delay may exceed the per-burst delay bound. In the following, we present our formulations for burst grooming in OBS networks to overcome the limitations of formulations of [93].

The objectives of the following 8 ILP formulations which correspond to the 8 grooming cases described in Section 3.2, respectively, are to minimize the number of formed larger bursts (i.e., maximize the number of original smaller-sized bursts groomed), and at the same time, to reduce the number of used wavelength converters (if wavelength conversion is available) and the number of used FDLs (if FDLs are available), that is, to strike a proper balance between burst grooming and grooming cost.

3.3.1 Notation

Notation used in the ILP formulations is defined as follows:

- $W$: number of wavelengths on a fiber;
- $N_i$: number of bursts on wavelength $i$;
• $a^n_i$: arriving time of burst $n$ on wavelength $i$;

• $e^n_i$: ending time of burst $n$ on wavelength $i$;

• $l^n_i$: length of burst $n$ on wavelength $i$;

• $T_x$: switching time;

• $T_{\text{max}}$: maximum length of the groomed bursts;

• $\delta$: FDL granularity;

• $D$: maximum time that a burst can be delayed by FDLs at a node;

• $\alpha$: cost of wavelength conversion;

• $\beta$: cost of one unit of FDL.

### 3.3.2 ILP1: Proportional Delay Slack Distribution with No Wavelength Conversion or FDLs

**Objective:**

$$\maximize \sum_{i=1}^{W} \sum_{n=1}^{N_i} x^n_i$$  \hspace{1cm} (3.1)

$$x^n_i = \begin{cases} 1 & \text{if bursts } n \text{ and } n - 1 \text{ groomed on wavelength } i; \\ 0 & \text{otherwise.} \end{cases}$$

$x^n_i \in \{0,1\}$ is a decision variable indicating whether burst $n$ on wavelength $i$ is groomed with its preceding burst or not.

**Subject to:**

$$g^n_i = a^n_i - e^{n-1}_i, \; \forall i \in \{1 \ldots W\}, \; \forall n \in \{1 \ldots N_i\}. \tag{3.2}$$
Eq. (3.2) defines the inter-burst gap between burst \( n \) and \( n - 1 \) on wavelength \( i \).

\[
\sum_{n=1}^{N_i} x^n_i(l^n_i + g^n_i) \leq T_{\text{max}}, \quad \forall i \in \{1 \ldots W\}. \tag{3.3}
\]

Eq. (3.3) is to ensure that the groomed burst does not exceed the maximum burst length.

\[
x^n_i \cdot d^n_i = 0, \quad \forall i \in \{1 \ldots W\}, \quad \forall n \in \{1 \ldots N_i\}. \tag{3.4}
\]

\[
d^n_i = \begin{cases} 
\left\lceil \frac{g^n_i - T_x}{\delta} \rightceil & g^n_i > T_x; \\
0 & \text{otherwise.} 
\end{cases}
\]

Eq. (3.4) is to ensure that two bursts are groomed together only if the gap between those two bursts \( g^n_i \leq T_x \).

### 3.3.3 ILP2: Proportional Delay Slack Distribution with Wavelength Conversion Only

**Objective:**

\[
\text{maximize} \sum_{i=1}^{W} \sum_{j=1}^{W} \sum_{k=1}^{W} \sum_{p=1}^{N_i} \sum_{q=1}^{N_j} \sum_{r=1}^{N_k} x^{p,q,r}_{i,j,k} - \alpha \sum_{i=1}^{W} \sum_{n=1}^{N_i} y^n_i \tag{3.5}
\]

\[
x^{p,q,r}_{i,j,k} = \begin{cases} 
1 & \text{if bursts } p \text{ and } q \text{ are groomed before burst } r; \\
0 & \text{otherwise.} 
\end{cases}
\]

\[
y^n_i = \begin{cases} 
1 & \text{on wavelength } i; \\
0 & \text{otherwise.} 
\end{cases}
\]

45
$x_{i,j,k}^{p,q,r} \in \{0,1\}$ is a decision variable indicating bursts $p$, $q$ and $r$ are on wavelengths $i$, $j$ and $k$ respectively. $y_i^p$ is also a decision variable indicating whether the wavelength of a burst is converted or not.

Subject to:

\[
\sum_{j=1}^{W} \sum_{k=1}^{N_j} \sum_{r=1}^{N_k} x_{i,j,k}^{p,q,r} \leq 1, \tag{3.6}
\]

$\forall i \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}$.

\[
\sum_{i=1}^{W} \sum_{k=1}^{N_i} \sum_{q=1}^{N_j} \sum_{r=1}^{N_k} x_{i,j,k}^{p,q,r} \leq 1, \tag{3.7}
\]

$\forall j \in \{1 \cdots W\}, \forall q \in \{1 \cdots N_j\}$.

\[
\sum_{k=1}^{W} \sum_{r=1}^{N_k} x_{i,j,k}^{p,q,r} \leq 1, \tag{3.8}
\]

$\forall i \in \{1 \cdots W\}, \forall j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}$.

Eq. (3.6) - (3.8) is to ensure that a burst can be groomed with only one burst and can be converted to only one wavelength.

\[
y_i^p = \sum_{j=1}^{W} \sum_{k \neq i}^{N_j} \sum_{q=1}^{N_k} x_{i,j,k}^{p,q,r} + \sum_{j=1}^{W} \sum_{k \neq i}^{N_j} \sum_{q=1}^{N_k} \sum_{r=1}^{N_k} x_{j,i,k}^{q,p,r}, \tag{3.9}
\]

$\forall i \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}$.

Eq. (3.9) is to ensure that the wavelength of a burst is converted to another wavelength only if this burst is groomed.

\[
x_{i,j,k}^{p,q,r} \cdot (g_k^r - (l_i^p + l_j^q + g_{i,j}^{p,q})) \geq 0, \forall i, j, k \in \{1 \cdots W\}, \tag{3.10}
\]

$\forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}, \forall r \in \{1 \cdots N_k\}$.

\[
g_k^r = a_k^r - e_k^{r-1}, \forall k \in \{1 \cdots W\}, \forall r \in \{1 \cdots N_k\}. \tag{3.11}
\]
\[ g_{i,j}^{p,q} = a_j^q - e_i^p, \quad (3.12) \]

\[ \forall i, j \in \{1 \ldots W\}, \forall p \in \{1 \ldots N_i\}, \forall q \in \{1 \ldots N_j\}. \]

\[ g_{i,j}^{p,q} > 0, \forall i, j \in \{1 \ldots W\}, \forall p \in \{1 \ldots N_i\}, \forall q \in \{1 \ldots N_j\}. \quad (3.13) \]

Eq. (3.10) to Eq. (3.13) are to ensure that a burst is groomed on a wavelength only if the grooming will not incur conflicts.

\[ x_{i,j,k}^{p,q,r} \cdot d_{i,j}^{p,q} = 0, \forall i, j, k \in \{1 \ldots W\}, \quad (3.14) \]

\[ \forall p \in \{1 \ldots N_i\}, \forall q \in \{1 \ldots N_j\}, \forall r \in \{1 \ldots N_k\}. \]

\[ d_{i,j}^{p,q} = \begin{cases} \left\lceil \frac{g_{i,j}^{p,q} - T_x}{\delta} \right\rceil & \text{if } g_{i,j}^{p,q} > T_x; \\ 0 & \text{otherwise}. \end{cases} \]

Eq. (3.14) is to ensure that two bursts are groomed together only if the gap between those two bursts \( g_{i,j}^{p,q} \leq T_x \).

\[ \sum_{p=1}^{N_i} \sum_{q=1}^{N_j} \sum_{r=1}^{N_k} x_{i,j,k}^{p,q,r} (t_j^q + g_{i,j}^{p,q}) \leq T_{\text{max}}, \forall i, j, k \in \{1 \ldots W\}. \quad (3.15) \]

Eq. (3.15) is to ensure that the groomed burst does not exceed the maximum burst length.

### 3.3.4 ILP3: Proportional Delay Slack Distribution with FDLs Only

**Objective:**

\[ \text{maximize } \sum_{i=1}^{W} \sum_{n=1}^{N_i} x_i^n - \beta \sum_{i=1}^{W} \sum_{n=1}^{N_i} (x_i^n \cdot d_i^n) \quad (3.16) \]
\[ x_i^n = \begin{cases} 1 & \text{if bursts } n \text{ and } n-1 \text{ groomed on wavelength } i; \\ 0 & \text{otherwise.} \end{cases} \]

\[ d_i^n = \begin{cases} \left\lceil \frac{g_i^n - T_x}{\delta} \right\rceil & g_i^n > T_x; \\ 0 & \text{otherwise.} \end{cases} \]

\[ g_i^n = a_i^n - e_i^{n-1}, \forall i \in \{1 \cdots W\}, \forall n \in \{1 \cdots N_i\}. \quad (3.17) \]

\( g_i^n \) is the inter-burst gap between burst \( n \) and \( n-1 \) on wavelength \( i \). \( x_i^n \in \{0, 1\} \) is a decision variable indicating whether burst \( n \) on wavelength \( i \) is groomed with its preceding burst or not.

Subject to:

\[ d_i^n \cdot \delta \leq \frac{d_s}{m}, \forall i \in \{1 \cdots W\}, \forall n \in \{1 \cdots N_i\}. \quad (3.18) \]

\[ d_i^n \cdot \delta \leq D, \forall i \in \{1 \cdots W\}, \forall n \in \{1 \cdots N_i\}. \quad (3.19) \]

Eq. (3.18) and Eq. (3.19) are to ensure that the delay of a burst does not exceed the delay bound.

\[ \sum_{n=1}^{N_i} x_i^n (l_i^n + \min(g_i^n, g_i^n - d_i^n \cdot \delta)) \leq T_{\max}, \forall i \in \{1 \cdots W\}. \quad (3.20) \]

Eq. (3.20) is to ensure that the groomed burst does not exceed the maximum burst length.

**3.3.5 ILP4: Proportional Delay Slack Distribution with Wavelength Conversion and FDLs**

Objective:

\[ \text{maximize} \sum_{i=1}^{W} \sum_{j=1}^{W} \sum_{k=1}^{N_i} \sum_{p=1}^{N_j} \sum_{q=1}^{N_k} \sum_{r=1}^{N_k} x_{i,j,k}^{p,q,r} - \alpha \sum_{i=1}^{W} \sum_{n=1}^{N_i} y_i^n \quad (3.21) \]
\[-\beta \sum_{i=1}^{W} \sum_{j=1}^{W} \sum_{k=1}^{N_i} \sum_{p=1}^{N_j} \sum_{q=1}^{N_k} \sum_{r=1}^{N} (x_{i,j,k}^{p,q,r} \cdot d_{i,j}^{p,q}) \]

\[x_{i,j,k}^{p,q,r} = \begin{cases} 
1 & \text{if bursts } p \text{ and } q \text{ groomed before burst } r; \\
0 & \text{otherwise.}
\end{cases} \]

\[y_{i}^{n} = \begin{cases} 
1 & \text{if wavelength converts when burst } n \text{ groomed on wavelength } i; \\
0 & \text{otherwise.}
\end{cases} \]

\[x_{i,j,k}^{p,q,r} \in \{0, 1\} \] is a decision variable indicating bursts \( p, q \) and \( r \) are on wavelengths \( i, j \) and \( k \) respectively. \( y_{i}^{n} \) is also a decision variable indicating whether the wavelength of a burst is converted or not.

\[d_{i,j}^{p,q} = \begin{cases} 
\left\lceil \frac{g_{i,j}^{p,q} - T_x}{\delta} \right\rceil & g_{i,j}^{p,q} > T_x; \\
0 & \text{otherwise.}
\end{cases} \]

\[g_{i,j}^{p,q} = a_j^q - e_i^p, \tag{3.22} \]

\[\forall i, j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}. \]

Subject to:

\[\sum_{j=1}^{W} \sum_{k=1}^{N_i} \sum_{q=1}^{N_j} \sum_{r=1}^{N} x_{i,j,k}^{p,q,r} \leq 1, \tag{3.23} \]

\[\forall i \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}. \]

\[\sum_{i=1}^{W} \sum_{k=1}^{N_i} \sum_{p=1}^{N_j} \sum_{r=1}^{N} x_{i,j,k}^{p,q,r} \leq 1, \tag{3.24} \]

\[\forall j \in \{1 \cdots W\}, \forall q \in \{1 \cdots N_j\}. \]
\[
\sum_{k=1}^{W} \sum_{r=1}^{N_k} x_{i,j,k}^{p,q,r} \leq 1,
\]
\[\forall i \in \{1 \cdots W\}, \forall j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}.
\]

Eq. (3.23) - (3.25) is to ensure that a burst can be groomed with only one burst and can be converted to only one wavelength.

\[
y_i^p = \sum_{j=1}^{W} \sum_{k \neq i}^{N_j} \sum_{q=1}^{N_k} x_{i,j,k}^{p,q,r} + \sum_{j=1}^{W} \sum_{k \neq i}^{N_j} \sum_{q=1}^{N_k} \sum_{r=1}^{N_k} x_{j,i,k}^{q,p,r},
\]
\[\forall i \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}.
\]

Eq. (3.26) is to ensure that the wavelength of a burst is converted to another wavelength only if this burst is groomed.

\[
d_{i,j}^{p,q} \cdot \delta \leq \frac{d}{m}, \forall i, j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}.
\]
\[\text{(3.27)}
\]

\[
d_{i,j}^{p,q} \cdot \delta \leq D, \forall i, j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}.
\]
\[\text{(3.28)}
\]

Eq. (3.27) and Eq. (3.28) are to ensure that the delay of a burst does not exceed the delay bound.

\[
x_{i,j,k}^{p,q,r} \cdot (g_k^r - (l_i^p + l_j^q + d_{i,j}^{p,q} - d_{i,j}^{p,q})) \geq 0, \forall i, j, k \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}, \forall r \in \{1 \cdots N_k\}.
\]
\[\text{(3.29)}
\]

\[
g_k^r = a_k^r - e_k^{r-1}, \forall k \in \{1 \cdots W\}, \forall r \in \{1 \cdots N_k\}.
\]
\[\text{(3.30)}
\]

\[
g_{i,j}^{p,q} > 0, \forall i, j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_i\}, \forall q \in \{1 \cdots N_j\}.
\]
\[\text{(3.31)}
\]
Eq. (3.29) to Eq. (3.31) are to ensure that a burst is groomed on a wavelength only if the grooming will not incur conflicts.

\[
\sum_{p=1}^{N_i} \sum_{q=1}^{N_j} \sum_{r=1}^{N_k} \sum_{i,j,k} x_{i,j,k}^{p,q,r} (l_{q}^{i,j} + g_{p,q}^{i,j} - d_{p,q}^{i,j}) \leq T_{\text{max}}, \forall i,j \in \{1 \cdots W\}. \tag{3.32}
\]

Eq. (3.32) is to ensure that the groomed burst does not exceed the maximum burst length.

3.3.6 ILP5: Collective Delay Slack Distribution with No Wavelength Conversion or FDLs

The same as case 1.

3.3.7 ILP6: Collective Delay Slack Distribution with Wavelength Conversion Only

The same as case 2.

3.3.8 ILP7: Collective Delay Slack Distribution with FDLs Only

Similar to Case 3, except that constraint Eq. (3.18) should be Eq. (3.33)

\[
d_{s}^{\delta} \leq d_{s}, \forall i \in \{1 \cdots W\}, \forall n \in \{1 \cdots N_{i}\}. \tag{3.33}
\]

3.3.9 ILP8: Collective Delay Slack Distribution with Wavelength Conversion and FDLs

Similar to Case 4, except that constraint Eq. (3.27) should be Eq. (3.34)

\[
d_{s}^{\delta} \leq d_{s}, \forall i,j \in \{1 \cdots W\}, \forall p \in \{1 \cdots N_{i}\}, \forall q \in \{1 \cdots N_{j}\}. \tag{3.34}
\]
3.3.10 Per-Hop Traffic Grooming Problem is NP-Complete

If the cost of wavelength conversion and FDLs is ignored and only one wavelength is considered, the per-hop traffic grooming problem is to form the smallest number of larger bursts on the wavelength. The per-hop traffic grooming problem can be easily shown to be \( \mathcal{NP} \)-complete by reducing the well-known Bin-Packing Problem. If all the wavelengths, wavelength conversion and FDLs are taken into account, the per-hop traffic grooming problem in OBS networks will be more complex. Therefore, the per-hop traffic grooming in general is also \( \mathcal{NP} \)-complete. In the next section, we propose efficient heuristic algorithms to solve the problem.

3.4 Heuristic Burst Grooming Algorithms

Additional notation used in the heuristic algorithms is defined as follows:

- \( b^n_i \): burst \( n \) on wavelength \( i \);
- \( B^n_i \): burst \( n \) on wavelength \( i \) formed by grooming;
- \( L^n_i \): length of burst \( B^n_i \);
- \( G^{p,q}_{i,j} \): inter-burst gap between bursts \( B^p_i \) and \( B^q_j \).

The Groom procedure (Fig. 3.4.1) is the main function for the per-hop traffic grooming algorithm. First, the Groom procedure executes the function Groom_SameWavelength (Fig. 3.4.2) to groom bursts on the same wavelength. Second, the Groom procedure tries to groom those bursts that are not groomed in the first step. Specifically, it uses two alternatives: (A1) use wavelength conversion and
then followed by using FDLs during grooming; (A2) use FDLs and then followed by using wavelength conversion during grooming. Finally, the Groom procedure executes function Groom_VoidFilling() (Fig. 3.4.7) to groom those bursts that are not groomed in the first two steps by using the voids on wavelengths. The second step and the third step require the availability of wavelength converters and FDLs.

Groom()
1. Groom_SameWavelength();
2. if wavelength conversion is available
3. for $w = 1$ to $W$ do
4. for each $B_{iw}$
5. if $|B_{iw}| = 1$ // $B_{iw}$ contains only one component burst
6. $\text{Time\_Limit} = \min(D, \text{delay bound of } b_{iw})$;
7. for $w' = 1$ to $W$ do
8. if Groom_WC_FDL($B_{iw}, \text{Time\_Limit}, w'$)
9. break;
10. endif
11. endfor
12. if $B_{iw}$ is not groomed
13. Groom_FDL_WC($B_{iw}, \text{Time\_Limit}, w$)
14. endif
15. endif
16. endfor
17. endfor
18. Groom_VoidFilling();
19. endif

Figure 3.4.1: Pseudo code for traffic grooming.

Function Groom_SameWavelength grooms the bursts starting from the latest burst on each wavelength. A burst will be groomed with its preceding burst(s) if the inter-burst gap criterion (line 14 Fig. 3.4.2) can be satisfied.

Function Groom_WC_FDL($B_{ni}, \text{Time\_Limit}, w$) (Fig. 3.4.3) grooms burst $B_{ni}$ with a burst on the specified wavelength $w$ starting from 0 delay within the delay time bound $\text{Time\_Limit}$ (equivalent to the OPG model of [97]).

In function Groom_OnWavelength (Fig. 3.4.4), there are 3 cases that burst $B_{pi}$ can be groomed with a burst on wavelength $w$ without any additional delay: (1) the
Groom_SameWavelength()
1. for w = 1 to W do
2. i = N_w; //start grooming from the latest burst on wavelength w
3. j = 0;
4. while i > 1 do
5. j = j + 1;
6. B^j_w = φ + b^j_w; //initialize each larger burst
7. L^j_w = l^i_w;
8. k = d^j_w;
9. Time_Limit = min(D, delay bound of b^j_w);
10. i' = i - 1; // try to groom each burst
11. if kδ ≤ Time_Limit
12. while i' ≥ 1 do
13. //burst b^i_w and b^i'_{w} can not be groomed together
14. if (g^i',w > T_{w}) or (l^i_w + l^i'_{w} + g^i',w - kδ > T_{max}) break;
15. else
16. B^i_w = B^j_w + b^i_w; //add burst b^i_w into the larger burst
17. L^i_w = l^i_w + l^i'_{w} + g^i',w - kδ;
18. i' = i' - 1; //continue to the next smaller burst
19. endif
20. endwhile
21. endif
22. i = i';
23. endwhile
24. endfor

Figure 3.4.2: Pseudo code for grooming bursts on the same wavelength.

burst arrives earlier than B^p_i and does not overlap with B^p_i in time; (2) the burst arrives later than B^p_i and does not overlap with B^p_i in time; or (3) the burst overlaps with B^p_i in time and there is enough void between the component bursts in B^q_w to accommodate B^p_i. In all the three cases, the inter-burst gap and burst length bound criteria should be satisfied.

Groom_WC_FDL(B^p_i, Time_Limit, w)
1. k = 0; // start with 0 delay
2. groomed = false;
3. while (kδ ≤ Time_Limit) and (not groomed) do
4. delay B^p_i by kδ;
5. groomed = Groom_OnWavelength(B^p_i, w);
6. k = k + 1;
7. endwhile
8. return groomed;

Figure 3.4.3: Pseudo code for grooming a burst with some bursts on a specified wavelength and within a specified delay bound.
Groom\_OnWavelength(B^p_i, w)
1. groomed = false;
2. for each B^q_w do
3. if Can\_Groom(B^p_i, B^q_w) and G^{w,q+1}_w > L_w
4. \( L^q_w = L^q_w + L^p_i + G^{p,q}_{i,w} \);
5. groomed = true;
6. break;
7. else
8. if Can\_Groom(B^p_i, B^q_w)
9. if \( G^{q-1,q}_w > L^q_i \)
10. \( L^q_w = L^q_w + L^p_i + G^{p,q}_{i,w} \);
11. groomed = true;
12. endif
13. else if B^q_w overlaps with B^p_i in time and there is enough void between the component bursts in B^q_w to accommodate B^p_i
14. groomed = true;
15. endif
16. endif
17. if groomed
18. B^q_w = B^q_w + B^p_i;
19. convert B^p_i to wavelength w;
20. break;
21. endif
22. endfor return groomed;

Figure 3.4.4: Pseudo code for grooming a burst with some bursts on a specified wavelength and without additional delay.

Function Can\_Groom(B^p_i, B^q_j) (Fig. 3.4.5) called in Groom\_OnWavelength decides whether two bursts can be groomed or not based on the inter-burst gap and burst length. If the inter-burst gap does not exceed \( T_x \) and the length of combined bursts and gap is not larger than \( T_{max} \), this function returns true; otherwise, it returns false.

Can\_Groom(B^p_i, B^q_j)
1. if \((0 \leq G^{p,q}_{i,j} \leq T_x)\) and \((L^p_i + L^q_j + G^{p,q}_{i,j} \leq T_{max})\)
2. return true
3. else return false;
4. endif

Figure 3.4.5: Pseudo code for deciding whether two large bursts can be groomed.

Function Groom\_FDL\_WC (Fig. 3.4.6) finishes the task of grooming first using FDLs followed by using wavelength conversion. Function Groom\_VoidFilling (Fig. 3.4.7) converts the wavelengths of two bursts to a third wavelength, making use
of the void on that wavelength to accommodate the groomed bursts with the aid of FDLs and trying to minimize the void on the wavelengths at the same time.

\[
Groom_{FDL,WC}(B^n_i, \text{Time\_Limit}, w)
\]
1. \( k = 0; \)
2. while \( k \delta \leq \text{Time\_Limit} \) do
3. delay \( B^n_i \) by \( k \delta; \)
4. for \( j = 1 \) to \( W \) do
5. if \( Groom\_On\_Wavelength(B^n_i, j) \)
6. \( w = j; \)
7. return true;
8. endif
9. \( k = k + 1; \)
10. endwhile
11. return false;

Figure 3.4.6: Pseudo code for finishing grooming first using FDLs followed by using wavelength conversion.

### 3.5 Performance Evaluation

#### A. Simulation setup

We evaluate the performance of our proposed ILP optimization models and heuristic algorithms. We have simulated a single OBS node using ns-obs version 0.6. In this case, there is no difference between proportional delay slack distribution and collective delay slack distribution. Several parameters may affect the grooming performance. The default parameters used in the simulations are as follows:

- Protocol: UDP;
- Bandwidth per wavelength: 2488Mbps;
- \( T_{max} \): 125\( \mu s; \)
- \( T_x \): 10\( \mu s; \)
Groom_VoidFilling() //groom two bursts on a third wavelength
1. for \( i = 1 \) to \( W \) do
2. for each ungroomed \( b^p_i \) do
3. \( r = \min(D, \text{delay bound of } b^p_i) \);
4. for \( j = 1 \) to \( W \) do
5. for each ungroomed \( b^q_j \) do
6. if \( g^{q,p}_{j,i} > 0 \)
7. if \((l^p_i + l^q_j + g^{q,p}_{j,i} - \delta \cdot d^{q,p}_{j,i} \leq T_{max})\) and \((\delta \cdot d^{q,p}_{j,i} \leq r)\)
8. delay burst \( b^p_i \) by \( \delta \cdot d^{q,p}_{j,i} \);
9. /*find the delay bound that these 2 bursts can be delayed at the same time*/
10. \( k = \min(\text{delay bound of } b^q_j, D - \delta \cdot d^{q,p}_{j,i}) \);
11. \( k = \frac{k}{\delta} \);
12. for \( l = 0 \) to \( k \) do
13. delay \( b^p_i \) and \( b^q_j \) by \( l \cdot \delta \);
14. find the wavelengths that have enough void to accommodate \( b^p_i \) and \( b^q_j \);
15. if several wavelengths are found
16. select the wavelength on which it causes the least void to groom burst \( b^p_i \) and \( b^q_j \);
17. endif
18. endif
19. endif
20. endif
21. endfor
22. endfor
23. endfor

Figure 3.4.7: Pseudo code for grooming two bursts on a third wavelength with void filling.

- Burst size: 5000 bytes;
- Packet size: 600;
- \( nCC \): each fiber comprises of 2 control wavelengths;
- \( nDC \): each fiber comprises of 3 data wavelengths;
- \( nFDL \): number of FDLs is 5;
- \( \delta \): FDL granularity is 10\( \mu s \);
- OBS scheduler: latest available unscheduled channel with void filling (LAUC-
Traffic generator: generates continuous traffic with exponential distribution of the packet inter-arrival time; the average packet inter-arrival time is determined by the packet size and offered load.

In the simulations, the grooming algorithm periodically processes control packets in batches every 2.5ms. For each periodic interval, the received control packets are sorted and processed in descending order of data burst arrival time. The simulations are run with 10 random seeds, each of which have 10 simulation runs. 95% confidence intervals are calculated.

Our heuristic is run after running the scheduler. For the non-grooming scenario, the bursts are scheduled only by the LAUC-VF scheduler. Full wavelength conversion ability is assumed. FDLs are available on a node. Grooming performance is measured by the ratio of the number of larger bursts obtained after grooming to the number of original smaller bursts before grooming. The simulation is run with different offered loads which are defined as the ratio of the used data wavelength bandwidth to the maximum data wavelength bandwidth.

**B. Comparison between ILP and Heuristic Algorithm**

In the ILP optimization model, we set both $\alpha$ and $\beta$ to 0.1. Table 3.1 shows the grooming performance comparison between ILP and heuristic for one specific burst set; Table 3.2 shows the average grooming performance comparison between ILP and heuristic for 100 burst sets. All the ILP optimization obtains the optimal solutions when the offered load is below 0.4. ILP grooming performance becomes better as the
Table 3.1: Grooming Performance Comparison between ILP and Heuristic for One Specific Burst Set (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$, $\alpha = 0.1$, $\beta = 0.1$).

<table>
<thead>
<tr>
<th>Load</th>
<th>ILP</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>82.2</td>
<td>82.2</td>
</tr>
<tr>
<td>0.2</td>
<td>33.7</td>
<td>41.6</td>
</tr>
<tr>
<td>0.3</td>
<td>27.2</td>
<td>36.0</td>
</tr>
<tr>
<td>0.4</td>
<td>20.2</td>
<td>27.7</td>
</tr>
</tbody>
</table>

Table 3.2: Average Grooming Performance Comparison between ILP and Heuristic for 100 Burst Sets (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$, $\alpha = 0.1$, $\beta = 0.1$).

<table>
<thead>
<tr>
<th>Load</th>
<th>ILP</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>83.0</td>
<td>83.0</td>
</tr>
<tr>
<td>0.2</td>
<td>34.4</td>
<td>40.9</td>
</tr>
<tr>
<td>0.3</td>
<td>25.6</td>
<td>35.0</td>
</tr>
<tr>
<td>0.4</td>
<td>16.5</td>
<td>30.5</td>
</tr>
</tbody>
</table>

offered load increases. The ILP performance is better than the heuristic algorithm and is close to that of the heuristic algorithm when the load is low. The grooming performance of our heuristic algorithm can get closer to that of ILP with more cost of wavelength conversion and FDLs.

C. Impact of Packet Size

Grooming performance versus various packet sizes with different network offered loads is shown in Fig. 3.5.8. We note that in general, grooming performance is better for higher traffic loads when packet size is fixed. For a fixed burst size, burst generation rate is affected by packet size and packet arrival rate. For a fixed burst size and a fixed packet size, a higher network offered load means a smaller packet inter-arrival time and more packets, which in turn will result in a smaller inter-burst gap and more bursts on the wavelengths. So more bursts can be groomed due to small inter-burst gap and by using FDLs. We also can see that a smaller packet size result
in a better grooming performance when the offered load is less than 0.7. An exception occurs at offered load 0.6 when the packet size is 200. This is because more bursts are dropped by the ns-obs scheduler. For a fixed offered load, a smaller packet size causes smaller packet inter-arrival time, which may also result in a smaller inter-burst gap and more bursts on the wavelengths. The network is so packed with bursts that the packet size has little impact on the grooming performance for offered loads above 0.7. Therefore, all the packet sizes achieve almost the same grooming performance when the offered load is higher than 0.7.

Figure 3.5.8: Grooming performance with various packet sizes ($T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$).

**D. Impact of Switching Time**

As for varying switching time, the simulation results in Fig. 3.5.9 show that the larger switching time is, the better grooming performance. As the switching time is increased from $5\mu s$ to $15\mu s$, the worst grooming performance is obtained at offered
load 1.0 where grooming performance grows from 23.5% to 16.3%, and at offered load 0.2, grooming performance grows from 66.8% to 34.8%. Although switching time is one factor that decides whether bursts can be groomed or not, it has no influence on packet generation. Thus, with a larger switching time, bursts are more likely to be groomed.

Figure 3.5.9: Grooming performance with various switching time (PacketSize = 600, \(nCC = 2\), \(nDC = 3\), \(nFDL = 5\), \(\delta = 10\) \(\mu s\)).

**E. Impact of Number of FDLs**

Next we compare grooming performance with different number of FDLs. Fig. 3.5.10 illustrates that the grooming performance is almost the same regardless of the number of FDLs. This is because the bursts are scheduled “evenly” on all the data wavelengths by the LAUC-VF scheduler with the help of FDLs. This scheduler makes the inter-arrival time between the bursts almost the same or being some units of FDLs. There are only some minor differences (not quite visible in the figure) between the grooming performances when the offered load is above 0.8. This
is because when the offered load is very high, the scheduler can schedule a little more bursts on the wavelengths by using more FDLs.

![Grooming performance with various number of FDLs](image)

**Figure 3.5.10**: Grooming performance with various number of FDLs (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $\delta = 10\mu s$).

**F. Impact of FDL Granularity**

Increasing FDL granularity may increase the inter-burst gap because the gap between the bursts are some FDLs units. From the previous analysis, we know that a smaller inter-burst gap will make it easy for bursts to be groomed. Therefore, in general, a larger FDL granularity will result in worse grooming performance, which is depicted in Fig. 3.5.11. We also can see that the grooming performance with an FDL granularity of $10\mu s$ is close to that with an FDL granularity of $5\mu s$. This is because the FDL granularities in these two cases are not larger than the switching time which is $10\mu s$ in the simulation, which makes it easier for FDLs to make the inter-burst gaps between some of the bursts within the reach of the switching time.
Figure 3.5.11: Grooming performance under various FDL granularities (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$).

**G. Impact of Number of Data Wavelengths**

The impact of number of data wavelengths on grooming performance is given in Fig. 3.5.12 with the number of control wavelengths on each fiber fixed at 2. We can see that more data wavelengths result in a better grooming performance with an offered load lower than 0.8. Especially a much better grooming performance can be obtained from more data wavelengths with a light offered load. This is because bursts are more evenly distributed on all the wavelengths and the inter-burst gap on each wavelength is smaller when there are more data wavelengths on each fiber. All the data wavelengths will be much more loaded with offered load above 0.8 so that the number of data wavelengths has little influence on the grooming performance.

**H. Impact of FDLs and Wavelength Conversion**

Finally, we compare the grooming performance under different combinations of
absence/presence of FDLs and wavelength conversion. Fig. 3.5.13 shows that our heuristic can achieve the best performance with both FDLs and wavelength conversion available in a node. For the cases that wavelength conversion is available, the grooming performance is better than their counterparts where wavelength conversion is not available because bursts on different wavelengths cannot be groomed. Also, the cases with FDLs can obtain better grooming performance than corresponding cases without FDLs due to the effects of FDLs. We note that for the case without FDLs or wavelength conversion, the grooming performance deteriorates rapidly when the offered load is above 0.4. This is because when the network is more loaded, the inter-burst gap on the same wavelength has a tendency to be larger due to the use of LAUC-VF without FDLs scheduling algorithm. At the same time, wavelength conversion cannot be performed to groom bursts.

Figure 3.5.12: Grooming performance with various number of data wavelengths (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nFDL = 5$, $\delta = 10\mu s$).
Figure 3.5.13: Grooming performance with different combinations of absence/presence of FDLs and wavelength conversion (PacketSize = 600, $T_x = 10\mu s$, $nCC = 2$, $nDC = 3$, $nFDL = 5$, $\delta = 10\mu s$).

3.6 Summary

In this chapter, we have studied the per-hop burst grooming problem where bursts with the same next hop may be groomed together. The burst grooming problems are also formulated as integer linear programs under various network configurations. We perform initial studies on a single node case. Both the ILP results and the simulation results show that the proposed heuristic algorithms are effective under varying system parameters.
Chapter 4

Exploring Node Light-Splitting Capability for Burst Grooming in Optical Burst Switched Networks

4.1 Introduction

The previous work presented in Chapter 3 studies the per-hop burst grooming scheme in OBS, where bursts with the same next hop may be groomed together in order to reduce the switching overhead. Note that high-speed OBS networks demand an efficient burst scheduling and grooming algorithm as the ultra-high data rate leaves little time for burst scheduling and resource reservation. Basically, the OBS grooming should be done in real time, which means that a proposed solution should have a low time complexity. However, per-hop burst grooming scheme may cause significant computation overhead due to performing the grooming procedure at each hop. Therefore, we try to propose a simpler grooming scheme, such that the processing delay incurred at the core nodes is reduced.

In this chapter, assuming burst grooming can only be realized at edge nodes, we study the burst grooming problem where sub-bursts originating from the same source
may be groomed together regardless of their destinations under certain conditions, that is, sub-bursts with different destinations may be aggregated together, and transmitted as a single burst. Furthermore, similar to the previous work, we assume that the path dependent end-to-end propagation delay experienced by a sub-burst is $d_p$, which is less than the end-to-end delay bound $D$. The delay slack of the sub-burst is $d_s = D - d_p$. When the sub-burst times out at the end of the delay slack, it must be transmitted in order to meet its delivery deadline. This sub-burst can be groomed with other sub-bursts currently at the edge node to form a large data burst. Sub-bursts can be groomed subject to the burst grooming criteria: (1) the length of the groomed burst should not exceed the maximum burst length $L_{max}$ since excessively long burst may increase the burst blocking probability in OBS networks; (2) the end-to-end delay bound of the sub-bursts is guaranteed; (3) the minimum length of a groomed burst should be at least $L_{min}$ (padding is applied if necessary).

To support the capability of grooming bursts destined to different destinations, previous work [98–100] adopted an approach that separates a groomed burst into sub-bursts and drops sub-bursts only at a destination egress node. The advantage of this approach is its simplicity. The drawback is that bursts are likely to experience a long delay as will be illustrated in subsequent sections. In this work, we explore the capability that core nodes can split incoming light signals to support multicast [114–119] to achieve more efficient burst grooming. Specifically, core nodes can transmit the groomed burst to multiple downstream nodes if the sub-bursts in the groomed burst have different destinations. The groomed burst will traverse a tree which spans the source and all the destinations of the sub-bursts in the groomed burst. The
destination egress nodes recognize, de-burstify, and drop the sub-bursts destined to these nodes, i.e., the sub-bursts destined to these egress nodes are removed from the groomed burst. At the same time, the remaining sub-bursts may be groomed with sub-bursts at these egress nodes subject to burst grooming criteria. We propose two effective burst grooming algorithms, (1) a no over-routing waste approach (NoORW); and (2) a minimum relative total resource ratio approach (MinRTRR). Our simulation results have shown that the proposed algorithms are effective in terms of the burst blocking probability, the average burst end-to-end delay, the number of sub-bursts per groomed burst, as well as the resource waste.

The rest of this chapter is organized as follows. Section 4.2 describes the burst grooming problem considered in this chapter and node architectures that support the burst grooming. Burst grooming algorithms are presented in Section 4.3. Section 4.4 reports the performance evaluation. This chapter concludes in Section 4.5.

4.2 Node Architecture and Burst Grooming

A. Node Architecture
Figure 4.2.2: An example of core node architecture.

The diagram of an example edge node architecture supporting burst grooming is shown in Fig. 4.2.1. Incoming data packets (e.g., IP packets) are aggregated into sub-bursts through the burst assembly unit. Burst grooming and scheduling are combined together because burst grooming and scheduling correlate closely at the edge nodes. Sub-bursts are groomed and scheduled with schemes to be presented momentarily, and the groomed bursts are then transmitted to the downstream nodes.

When a burst reaches an egress edge node, de-grooming will first be performed to separate individual sub-bursts. Those sub-bursts destined to this edge node will be dropped and sent to the data sink for further processing, such as IP routing, while those sub-bursts which need to be removed by this edge node will be dropped and discarded as explained later. Burst grooming and scheduling will be performed again on the remainder of the groomed burst with sub-bursts at this edge node.

Fig. 4.2.2 shows a possible architecture of a core node that provides light splitting ability. The arriving input signals to a core node are de-multiplexed into separate
B. Burst Grooming Exploring Node Light Splitting Capability

We illustrate the burst grooming concept considered in this work using an example (Fig. 4.2.3).

**Example 1** Nodes 1, 3, 4, and 5 are edge nodes while node 2 is a core node. Sub-bursts $b_1$ and $b_2$ individually do not satisfy the minimum burst length requirement. The combined length of sub-bursts $b_1$ and $b_2$ is larger than the minimum burst length. Sub-bursts $b_1$ and $b_2$ have the same source and are destined to nodes 3 and 5, respectively. Assume $b_1$ times out at the end of its delay slack and that these two sub-bursts satisfy the burst grooming criteria. Source node 1 can groom these two sub-bursts and transmit them as a single data burst. The groomed burst will go through a tree that spans the source node 1, destination nodes 3 and 5. Core node 2 provides the light
splitting ability so that the groomed burst gets replicated at node 2 and forwarded to downstream nodes. Edge node 4 discards \( b_1 \) because node 3 is not reachable via node 4 and \( b_1 \) has been sent to node 3 through another tree branch. The transmission of \( b_1 \) over link (2,4) constitutes a waste of resource. \( b_2 \), the remainder of the groomed burst, can be groomed with sub-burst \( b_3 \) at node 4 that is destined to node 5. The newly groomed burst is transmitted to the downstream node 5. Destination node 3 recognizes and de-burstifies sub-burst \( b_1 \) and discards sub-burst \( b_2 \) while destination node 5 recognizes and de-burstifies sub-bursts \( b_2 \) and \( b_3 \).

\[ \square \]

**C. Burst Grooming Problem**

In this work, burst grooming considers a number of issues: (1) the length of a groomed burst should be at least \( L_{min} \) and is limited by the maximum burst length \( L_{max} \); (2) the end-to-end delay bound should be guaranteed for sub-bursts with delay requirements, and routing of a groomed burst should satisfy the end-to-end delay bound of these sub-bursts contained in the groomed burst; (3) an ingress node needs to decide how to aggregate sub-bursts, i.e., decide which sub-bursts should be groomed together; (4) an ingress node needs to find a route for each groomed burst; and (5) each core node needs to transmit a groomed burst to several downstream nodes.

Given a network \( G = (V, E) \) where \( V \) is the set of nodes and \( E \) is the set of links, for a sub-burst \( b_1 \) that times out at the end of its delay slack, the burst grooming problem is to find a set of sub-bursts which can be groomed with sub-burst \( b_1 \), and to determine a proper route to transmit the groomed burst such that total network
resources used and/or wasted (to be defined subsequently) for delivering the sub-
bursts is minimized.

4.3 Proposed Burst Grooming Algorithms

Notation used in this work is defined as follows:

- $b_i$: a sub-burst;

- $S = \{b_1, b_2, b_3, \ldots \}$: a set of sub-bursts at the same ingress node where $b_1, b_2, b_3, \ldots$ are sub-bursts;

- $B_i = \{b_1, \ldots \}$: a groomed burst which includes at least one sub-burst $b_1$ that times out first;

- $c_e$: the cost of using link $e$, taking on a positive value;

- $L_{b_i}$: the length of sub-burst $b_i$;

- $L_{B_i}$: the length of groomed burst $B_i$;

- $\Gamma_{b_i}$: the set of edges on a shortest path traversed by sub-burst $b_i$ without burst grooming;

- $\Upsilon_{B_i}$: the set of edges on the path traversed by the groomed burst $B_i$. In general, the path taken by the groomed burst can be a tree;

- $\psi_{b_i}$: the padding waste (if any) of sub-burst $b_i$ transmitted individually without grooming;

- $\psi_{B_i}$: the padding waste (if any) of groomed burst $B_i$;
• \( B^e_i \): a set of sub-bursts that go through link \( e \) and cause over-routing waste (to be defined below);

• \( \ell^e_i \): the total burst length of the sub-bursts in \( B^e_i \);

• \( \chi^e_i \): the over-routing waste incurred for \( B_i \) on link \( e \);

• \( \phi_B \): the total resources needed for delivering groomed burst \( B_i \) using route \( \Upsilon_B \);

• \( \varphi_B \): the total resources needed for delivering sub-bursts in \( B_i \) using a shortest path individually.

Without burst grooming, if the length of sub-burst \( b_i, L_{b_i} \), is less than \( L_{min} \), the sub-burst will be padded to \( L_{min} \). This incurs padding waste. Suppose the sub-burst traverses a shortest path \( \Gamma_{b_i} \) to its destination. The total padding waste, \( \psi_{b_i} \), is given by

\[
\psi_{b_i} = \sum_{e \in \Gamma_{b_i}} \max(L_{min} - L_{b_i}, 0) \cdot c_e. \tag{4.1}
\]

A groomed burst \( B_i \) still may not satisfy the minimum burst length requirement. Suppose the groomed burst is routed using path \( \Upsilon_{B_i} \). The padding waste of a groomed burst \( B_i, \psi_{B_i} \), is given by

\[
\psi_{B_i} = \sum_{e \in \Upsilon_{B_i}} \max(L_{min} - L_{B_i}, 0) \cdot c_e. \tag{4.2}
\]

The ultimate goal of burst grooming is to efficiently deliver bursts to their intended destinations and, at the same time, the delivery should not be over done, namely, sub-bursts should not traverse a link unnecessarily without contributing to the delivery of sub-bursts of \( B_i \) to their destinations using route \( \Upsilon_{B_i} \). Otherwise, over-routing waste
occurs. Note that in Example 1, for all sub-bursts to be delivered to their intended destinations, sub-burst $b_1$ does not need to go through link $(2, 4)$, and sub-burst $b_2$ does not need to go through link $(2, 3)$. Given a link $e$ that a groomed burst $B_i$ goes through, we define the set of sub-bursts that go through the link and cause over-routing waste as $B^e_i$, and the total burst length of the sub-bursts in $B^e_i$ is

$$t^e_{B_i} = \sum_{b \in B^e_i} L_b. \quad (4.3)$$

If $L_{B_i} < L_{\text{min}}$, $B_i$ has to be padded to $L_{\text{min}}$ before transmission. If the sub-bursts were to be delivered individually, each sub-burst in $B_i$ would have to be padded to satisfy the minimum burst. Because the length of the groomed burst is less than $L_{\text{min}}$, sub-bursts in $B_i$ can be treated as padding waste, i.e., there is no over-routing waste for $B_i$. In this case, we only consider padding waste of $B_i$. If $L_{B_i} - t^e_{B_i} \geq L_{\text{min}}$, the over-routing for $B_i$ on link $e$ is $t^e_{B_i} \cdot c_e$, because $B_i \setminus B^e_i$ satisfies the minimum burst length requirement. When $L_{B_i} - t^e_{B_i} \leq L_{\text{min}}$ and $L_{B_i} > L_{\text{min}}$, $B_i$ satisfies the minimum burst length requirement while $B_i \setminus B^e_i$ does not. The over-routing waste in this case is $(L_{B_i} - L_{\text{min}}) \cdot c_e$. Therefore, the over-routing waste incurred for $B_i$ on link $e$ is

$$\chi^e_{B_i} = \begin{cases} 
  t^e_{B_i} \cdot c_e & L_{B_i} \geq L_{\text{min}} + t^e_{B_i}; \\
  (L_{B_i} - L_{\text{min}}) \cdot c_e & L_{\text{min}} < L_{B_i} < L_{\text{min}} + t^e_{B_i}; \\
  0 & L_{B_i} \leq L_{\text{min}}.
\end{cases} \quad (4.4)$$

Eq. (4.4) can be simplified as

$$\chi^e_{B_i} = \min(t^e_{B_i}, \max(L_{B_i} - L_{\text{min}}, 0)) \cdot c_e. \quad (4.5)$$
The total over-routing waste of a groomed burst $B_i$ is

$$\chi_{B_i} = \sum_{e \in \Upsilon_{B_i}} \chi_{B_i}^e,$$  \hspace{1cm} (4.6)

assuming the groomed burst is routed using $\Upsilon_{B_i}$. On the other hand, the total resources needed for delivering a groomed burst $B_i$ using route $\Upsilon_{B_i}$ is given by

$$\phi_{B_i} = \sum_{e \in \Upsilon_{B_i}} \max(L_{\text{min}}, L_{B_i}) \cdot c_e.$$  \hspace{1cm} (4.7)

If the sub-bursts in the groomed burst $B_i$ are transmitted individually using a shortest path $\Gamma_{b_i}$, the total resources needed are

$$\varphi_{B_i} = \sum_{b_i \in B_i} \sum_{e \in \Gamma_{b_i}} \max(L_{\text{min}}, L_{b_i}) \cdot c_e.$$  \hspace{1cm} (4.8)

Sub-bursts can be groomed and routed in different ways. The burst grooming problem must decide how to construct a groomed burst from a set of available sub-bursts, $S$, based on the burst grooming criteria, as well as decide how to route the groomed burst. Intuitively, sub-bursts from $S$ can be groomed as long as they satisfy the burst grooming criteria. However, the resulting groomed burst may incur much resource waste and use more resources than necessary, increasing the burst blocking probability in the network. Therefore, care must be taken in designing burst grooming and routing algorithms. We propose two types of burst grooming and routing algorithms: (1) a no over-routing waste approach ($\text{NoORW}$); and (2) a minimum relative total resource ratio approach ($\text{MinRTRR}$).

A. **No Over-routing Waste (NoORW)**
In this approach, we require that burst grooming does not result in any over-routing waste, i.e.,

\[ \chi_{B_i} = 0, \quad (4.9) \]

or

\[ \sum_{e \in \Upsilon_{B_i}} \min(\iota_{B_i}, \max(L_{B_i} - L_{\text{min}}, 0)) \cdot c_e = 0, \quad (4.10) \]

where sub-bursts in the groomed burst is transmitted using a shortest path.

The rationale behind this approach is to ensure that burst delivery via burst grooming is as resource efficient as burst delivery without burst grooming. Basically, sub-bursts are groomed only if the burst grooming incurs no over-routing waste. For example, two or more sub-bursts can be groomed into a burst \( B_i \) when the longest shortest path used by a sub-burst in the groomed burst \( B_i \) uses all the links of other shortest paths taken by other sub-bursts in the groomed burst \( B_i \); or when \( L_{B_i} \leq L_{\text{min}} \), over-routing waste is considered to be zero.

**Example 2** In the network of Fig. 4.2.3, assume that two sub-bursts \( b_1 \) and \( b_2 \) originate from node 1, and are destined to nodes 3 and 4, respectively. Assume further that these two bursts satisfy the burst grooming criteria. Sub-burst \( b_1 \) times out at the end of its delay slack.

If \( L_{b_1} + L_{b_2} \leq L_{\text{Min}} \), these two sub-bursts can be groomed because the grooming will not incur any over-routing waste. However, if the schemes proposed in [98–100] are used, these two sub-bursts cannot be groomed based on the burst grooming criteria proposed. On the other hand, if \( L_{b_1} + L_{b_2} > L_{\text{Min}} \), these two sub-bursts cannot be groomed because burst grooming will result in some over-routing waste. In this case, if
and \( b_1 \) and \( b_2 \) are destined to nodes 5 and 4, respectively, these two bursts can be groomed as a result of no over-routing waste because the shortest path of \( b_1 \) uses all the links on the shortest path of \( b_2 \). The groomed burst will be sent to node 4 and then to node 5. However, the schemes proposed in [98–100] will result in the groomed burst being sent to node 5 and then to node 4, which increases the end-to-end delay. 

\[ \square \]

**B. Burst Grooming Problem with NoORW is NP-Complete**

**Theorem 4.1** Given a set of sub-bursts \( S = \{b_1, b_2, b_3, \ldots \} \) where \( b_1 \) is the sub-burst that times out, burst grooming problem with no over-routing waste is NP-complete.

**Proof:** Two cases need to be considered in this problem. In the first case, the longest shortest path used by a sub-burst in the groomed burst \( B_i \) uses all the links on other shortest paths of the sub-bursts in the groomed burst \( B_i \); while in the second case, there is no sub-burst whose shortest path covers all the links used by all the other shortest paths of the sub-bursts in the groomed burst \( B_i \). These two cases are similar except that the length of the groomed burst in the second case \( L_{B_i} \leq L_{\text{min}} \) before padding. We only need to prove that one of the cases is NP-complete. Without loss of generality, we prove that the second case is NP-complete. Next we show that the *Knapsack Problem*, a well-known NP-complete problem, can be reduced to the second case of the burst grooming problem with no over-routing waste.

Since \( b_1 \) must be in the groomed burst, we need to select other sub-bursts in \( S \), so that the length of the groomed burst does not exceed \( L_{\text{min}} \). The polynomial-time
reduction [120, 121] from one instance of the Knapsack Problem to one instance of the burst grooming problem with NoORW is obvious: the input set \( A \) in the Knapsack Problem is equivalent to the set of sub-bursts, \( S \), in the burst grooming problem; the size function and value function in the Knapsack Problem are essentially the same as the length function of the sub-bursts (i.e., the length of the sub-bursts) in the burst grooming problem; and the desired value is in the range of \( (0, L_{\text{min}}) \) and should be maximized. By this reduction, we can easily see that there is a solution to one instance of the burst grooming problem if and only if there is a solution to one instance of the Knapsack Problem. Hence, the burst grooming problem with NoORW is \( \mathcal{NP} \)-complete.

We propose a heuristic algorithm to solve the burst grooming problem with NoORW.

**C. Heuristic Algorithm for Burst Grooming Problem with NoORW**

Given a set of sub-bursts \( S \) in which the sub-burst \( b_1 \) times out and all the sub-bursts are transmitted along their shortest paths, the algorithm NoORWAAlg shown in Fig. 4.3.4 tries to groom \( b_1 \) with proper sub-bursts in the set \( S \) which is an input parameter. \( \alpha \) (\( 0 < \alpha < 1 \)) is another input parameter to the algorithm NoORWAAlg for controlling the precision of the final solution obtained with respect to the optimal solution, which will be explained momentarily.

Two cases are considered based on whether the longest shortest path of a sub-burst uses all the links on the shortest paths of other sub-bursts in the groomed burst. Note that \( b_1 \) will always be in the groomed burst. Line 1 deals with the case when
NoORWA_{Alg}(S, \alpha) 
// Case 1 
1. \(B = GetMaxLenSet(S \setminus \{b_1\}, L_{\min} - L_{b_1}, \alpha)\); 
// Case 2 
2. Find \(S'\), a subset of \(S\), such that \(S'\) includes \(b_1\), and the longest shortest path used by a sub-burst in \(S'\) uses all the links on the shortest paths of other sub-bursts in \(S'\); 
3. \(B' = GetMaxLenSet(S' \setminus \{b_1\}, L_{\max} - L_{b_1}, \alpha)\); 
4. if \(L_{B'} \geq L_B\) 
5. \(B' = B' \cup \{b_1\}\); 
6. Route \(B'\) using a shortest path tree; 
7. else \(B = B \cup \{b_1\}\) 
8. Route \(B\) using a shortest path tree; 

Figure 4.3.4: Pseudo code for burst grooming with no over-routing waste.

sub-bursts with different destinations are groomed. Lines 2-3 are similar to line 1, except that lines 2-3 deal with the case when the shortest path of a sub-burst uses all the links on the shortest paths of other sub-bursts. This algorithm \(NoORWA_{Alg}\) results in two groomed bursts \(B\) and \(B'\), and chooses the larger one in length between \(B\) and \(B'\) (lines 4-8) as the final groomed burst.

The sub-routine \(GetMaxLenSet\), used in the algorithm \(NoORWA_{Alg}\) and shown in Fig. 4.3.5, is to obtain a set of sub-bursts \(B (B \subseteq S)\) such that the total length of the sub-bursts in \(B\) is as large as possible but not larger than the length constraint \(L\), where \(S\), \(L\) and \(\alpha\) are input parameters. Given \(S = \{b_1, b_2, b_3, \ldots\}\), \(GetMaxLenSet\) examines all the sub-sets of \(S\), and trims the sub-sets by a trimming parameter \(\frac{\alpha}{|S|}\) as well as the length constraint \(L\). Lines 7-8 can be performed by a merge sort algorithm which runs in \(O(|Q| + |Q'|)\) time, since the elements in \(Q\) and \(Q'\) are sorted in non-decreasing order.

The trimming algorithm \(TrimSet\) shown in Fig. 4.3.6 checks each element in the sorted input set \(Q\) which contains the length values of groomed sub-bursts, and
GetMaxLenSet($S, L, \alpha$)

// $L$: the length constraint of a groomed burst
// $\alpha$: a constant
// $Q$: a set containing lengths of groomed sub-bursts

1. $Q = \emptyset$
2. for $i = 1$ to $|S|$
3. $Q' = Q$
4. if $L_{b_i} < L$
5. if $Q'$ is empty, add $L_{b_i}$ to $Q'$:
6. else increase each element in $Q'$ by $L_{b_i}$;
7. $Q = Q \cup Q'$;
8. sort $Q$ in non-decreasing order;
9. $Q = TrimSet(Q, L, \alpha |S|)$;
10. $B$ = the set of sub-bursts corresponding to the largest valued element in $Q$;
11. return $B$;

Figure 4.3.5: Pseudo code for obtaining a set of sub-bursts that have the largest total burst length.

TrimSet($Q, L, \beta$)

// $Q$: a set of length values of groomed sub-bursts
// $L$: a length constraint
// $\beta$: trim factor

1. $a = q_1 \in Q$
2. for $i = 2$ to $|Q|$
3. if $q_i > L$
4. remove $q_i$ from $Q$
5. else if $a \geq (1 - \beta) \cdot q_i$
6. remove $q_i$ from $Q$
7. else $a = q_i$
8. return $Q$

Figure 4.3.6: Pseudo code for trimming a set.
removes elements larger than the length constraint $L$. The algorithm also removes elements from $Q$ whose values are very close to each other by a factor of $1 - \beta$ (line 5). The elements in $Q = \{q_1, q_2, q_3, \ldots \}$ after trimming cannot be larger than $L$, and for each element $q_j$ removed from $Q$, there is an element $q_i$ remaining in $Q$, such that 
\[ \frac{q_j - q_i}{q_j} \leq \beta \] (lines 3-7). That is, for any two adjacent elements $q_i$ and $q_{i+1}$ remained in $Q$ after trimming

\[ q_i < (1 - \beta) \cdot q_{i+1} \leq L. \]  

(4.11)

The relative error introduced with respect to $q_i$ and $q_j$ when $q_j$ is removed is at most a factor of $1 - \beta$. This algorithm TrimSet runs in $O(|Q|)$ time. Algorithm GetMaxLenSet plays a very important role in finding a set of sub-bursts to be groomed with the timed-out burst $b_1$ in algorithm NoORWAlg, so that the length of the groomed burst is as large as possible but is within a length constraint $L$.

**Theorem 4.2** Given a set of sub-bursts $S = \{b_1, b_2, b_3, \ldots \}$, a length constraint $L$, and a precision parameter $\alpha$, algorithm GetMaxLenSet is a $(1 - \alpha)$-approximation algorithm.

**Proof:** Algorithm GetMaxLenSet tries to find all the sub-burst sets, some of which are removed by algorithm TrimSet. The elements that survive TrimSet must differ at least by a factor of $1 - \frac{\alpha}{|S|}$. According to Eq. (4.11), for any two adjacent elements $q_i$ and $q_{i+1}$ that are left in $Q$ in algorithm TrimSet

\[ q_i < (1 - \frac{\alpha}{|S|}) \cdot q_{i+1}. \]  

(4.12)

81
Thus,
\[|Q| \leq \log \frac{1}{1 - \frac{\alpha}{|S|}} L = \frac{\ln L}{-\ln (1 - \frac{\alpha}{|S|})}. \tag{4.13}\]

Because
\[
\ln (1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots. \tag{4.14}
\]
we have,
\[|Q| \leq \log \frac{1}{1 - \frac{\alpha}{|S|}} L = \frac{\ln L}{-\ln (1 - \frac{\alpha}{|S|})} < \frac{|S| \ln L}{\alpha}. \tag{4.15}\]

Therefore, the total number of elements remained after trimming cannot exceed \(\frac{|S| \ln L}{\alpha}\), which is polynomial in \(|S|\), and the length constraint \(L\) as well as the precision parameter \(\alpha\). Hence, \(GetMaxLenSet\) can be performed in polynomial time, and its time complexity is \(O(\frac{|S|^2 \ln L}{\alpha})\). The time complexity of \(NoORWAlg\) is then \(O(\frac{|S|^2 \ln L_{max}}{\alpha})\).

In each trimming step, the introduced relative error between the removed elements and the elements remained is at most a factor of \(1 - \frac{\alpha}{|S|}\). Therefore, the multiplicative error after the \(|S|\) trimming steps is at most a factor of \((1 - \frac{\alpha}{|S|})^{|S|}\).

Note that
\[
\frac{d}{dx}(1 - \frac{\alpha}{x})^x = x \cdot (1 - \frac{\alpha}{x})^{x-1} \cdot \frac{\alpha}{x^2} > 0, \tag{4.16}
\]
when \(x \geq 1\), which means that the function \((1 - \frac{\alpha}{x})^x\) increases as \(x\) \((x \geq 1)\) increases.

Since \(|S| \geq 1\), we have
\[
(1 - \frac{\alpha}{|S|})^{|S|} \geq 1 - \alpha. \tag{4.17}
\]
That is, assuming that the optimal solution results a groomed burst with a length of \(q^*\), \(GetMaxLenSet\) can obtain a solution \(q\) with the largest groomed burst length,
such that

\[(1 - \frac{\alpha}{|S|})|S|q^* \leq q, \tag{4.18}\]

\[(1 - \alpha) \cdot q^* \leq q \leq q^*. \tag{4.19}\]

Hence, algorithm GetMaxLenSet is a \((1 - \alpha)\)-approximation algorithm.

\[\square\]

**D. Minimum Relative Total Resource Ratio (MinRTRR)**

The no over-routing waste requirement is somewhat restrictive for burst grooming because each sub-burst needs to be transmitted on its shortest path. At the same time, the groomed burst may still need to be padded if its length is less than \(L_{\text{min}}\).

Note that the overall resources needed for burst delivery may be reduced with proper burst grooming even if the grooming process may incur some over-routing waste.

Therefore, we consider a second approach that reduces the resources needed in burst grooming compared against the case in which sub-bursts are delivered individually without grooming. In this case, the sub-bursts may not necessarily be transmitted using their shortest paths. The burst grooming performance is measured by a metric termed as the *relative total resource ratio* \((R)\) defined for a groomed burst \(B_i\):

\[R = \frac{\phi_{B_i}}{\varphi_{B_i}}, \tag{4.20}\]

or

\[R = \frac{\sum_{e \in \Upsilon_{B_i}} \max(L_{\text{min}}, L_{B_i}) \cdot c_e}{\sum_{b_i \in B_i} \sum_{e \in \Gamma_{b_i}} \max(L_{\text{min}}, L_{b_i}) \cdot c_e}. \tag{4.21}\]

The goal is to design burst grooming algorithms that minimize \(R\) and \(R \leq 1\), which means that proper burst grooming guarantees savings in network resource
used for burst delivery. If no burst grooming is performed, $B_i$ consists of only one sub-burst, in which case $R = 1$.

**Example 3** In the network of Fig. 4.2.3 where each link is assumed to have a cost of 1, two sub-bursts $b_1$ and $b_2$ originate from node 1 and are destined to nodes 3 and 4, respectively. Suppose that these two sub-bursts satisfy the burst grooming criteria.

If $L_{b_1} + L_{b_2} > L_{\text{Min}}$ (e.g., $L_{b_1} = 0.5 \cdot L_{\text{Min}}$ and $L_{b_2} = 0.6 \cdot L_{\text{Min}}$), these two sub-bursts cannot be groomed to ensure no over-routing waste. If $b_1$ and $b_2$ are not groomed, the total resources needed for their delivery is $4 \cdot L_{\text{min}}$. However, total resources needed would be $3.3 \cdot L_{\text{min}}$ if $b_1$ and $b_2$ are delivered in a groomed burst. This shows the benefit brought about by careful burst grooming.

Given a set of sub-bursts $S$ sorted by the sub-burst delivery deadline, when sub-burst $b_1$ times out, the grooming algorithm $\text{MinRTRRA}lg$ (shown in Fig. 4.3.7) determines the sub-bursts that should be groomed and a proper route for delivery. As a result, the algorithm obtains the groomed burst $B$, and the corresponding minimum $R$ whose value is indicated by $R_{\text{min}}$. Before the iterative search process, this algorithm initializes the groomed burst and the loop control parameter. Line 1 initializes the groomed burst $B$, which puts only one sub-burst $b_1$ into $B$, and initializes the edge set of the routing tree of $B$ and the burst length of $B$. Line 2 initializes the minimum $R$ as 1, since the maximum $R$ is 1. At the same time, the outermost loop control parameter is set to true to allow this algorithm to search for sub-bursts to be groomed into $B$. The innermost for loop (lines 7-14) finds the minimum $R$ for
MinRTRRAlg($S$)
1. $B = \{b_1\}; \ U_B = \Gamma_{b_1}; L_B = L_{b_1}$;
2. $R_{min} = 1; \ updated = \text{true}$;
3. while updated do
4. \hspace{0.5cm} $k = 0$;
5. \hspace{0.5cm} for $i = 2$ to $|S|$ \\
6. \hspace{1.0cm} if $L_B + L_{b_i} \leq L_{max}$ \\
7. \hspace{1.5cm} for each node $j$ that $B$ needs to go through \\
8. \hspace{2.0cm} assume $P'$ is the route of sub-burst $b_i$, \\
\hspace{2.5cm} such that $b_i$ first goes to node $j$ via $U_B$ and then \\
\hspace{3.0cm} goes to its destination through the shortest path from node $j$; \\
9. \hspace{1.5cm} if the end-to-end delay bound of $b_i$ is satisfied by going through $P'$ \\
10. \hspace{2.0cm} $B' = B \cup \{b_i\}; \ U_{B'} = U_B \cup P'; L_{B'} = L_B + L_{b_i}$;
11. \hspace{1.0cm} compute $R$ for $B'$; \\
12. \hspace{1.0cm} if $R < R_{min}$ \\
13. \hspace{1.5cm} $R_{min} = R; \ k = i; \ P = P'$;
14. \hspace{1.0cm} endfor
15. \hspace{0.5cm} endfor
16. if $k = 0$ then \\
17. \hspace{1.0cm} updated = false \\
18. \hspace{1.0cm} else \\
19. \hspace{1.5cm} $B = B \cup \{b_k\}$; \\
20. \hspace{1.5cm} $U_B = U_B \cup P; L_B = L_B + L_{b_k}$; \\
21. \hspace{1.5cm} $S = S \setminus \{b_k\}$; \\
22. \hspace{1.0cm} endwhile
23. return $B$;

Figure 4.3.7: Pseudo code for burst grooming that minimizes the relative total resource ratio.
a sub-burst \(b_i\) by trying to transmit \(b_i\) on different routes, if \(b_i\) and \(B\) satisfy the grooming criteria (lines 6 and 9). Specifically, assuming that \(b_i\) goes to node \(j\) (where node \(j\) is a node on \(\Upsilon_B\)) with \(B\) via the existing routing tree \(\Upsilon_B\) and then \(b_i\) is transmitted through the shortest path from node \(j\) to \(b_i\)’s destination, the algorithm can calculate the corresponding \(R\). The sub-burst \(b_k\) that results in the minimum \(R\) among all the sub-bursts can be found after the second for loop. The outermost while iteration (lines 3-22) checks whether more sub-bursts can be groomed into \(B\) to make \(R_{min}\) smaller. If sub-burst \(b_k\) that was found in the second for loop can result in a smaller \(R\) than the current one, it is groomed into \(B\) and removed from \(S\), and the corresponding information of the groomed burst \(B\) is updated (line 19-21); otherwise, the algorithm terminates the search process by setting the outermost loop control parameter to false (line 17), because the minimum \(R\) has been found.

4.4 Performance Evaluation

A. Simulation Setup

![14-node NSF network topology](image)

Figure 4.4.8: 14-node NSF network topology.

We evaluate the performance of our proposed heuristic algorithms by implement-
ing the algorithms using ns-obs version 0.9a [122]. Three network topologies are used in the simulation, named NSF network topology (Fig. 4.4.8), 16-node 30-edge network topology (Fig. 4.4.9), and 40-node 70-edge network topology (Fig. 4.4.10). In these three networks mentioned above, each link is bidirectional with a fiber in each direction. The number of data wavelengths and control wavelengths on each link are 8 and 2, respectively. The bandwidth of each wavelength is 10 Gbps. The minimum burst length requirement ($L_{\text{min}}$) is 5000 bytes, and the maximum burst length ($L_{\text{max}}$) is 40000 bytes. The average sub-burst size is 576 bytes, and the incoming self-similar traffic generated by OBS traffic generator is uniformly distributed between all pairs of edge nodes. The offset time for each sub-burst is 50 ms. The simulation is run till 95% confidence level.

In the simulation, bursts are scheduled with the latest available unscheduled channel with void filling (LAUC-VF) algorithm [53]. The grooming performance is mea-
sured by five metrics: burst blocking probability, average burst end-to-end delay, number of sub-bursts per groomed burst, padding waste, and over-routing waste. The simulation is run with different offered loads which are defined as the ratio of the offered traffic rate to the maximum data wavelength channel capacity. We compare the performance of both the NoORW approach and the MinRTRR approach with the case when no grooming is applied. The trimming parameter ($\alpha$) in the NoORW case is set to be 0.2 in the simulation.

**B. Burst Blocking Probability**

The burst blocking probability performance versus different offered loads is shown in Figs. 4.4.11 to 4.4.13. We notice that in general, burst grooming results in lower burst blocking probability than the case without grooming, because several sub-bursts that do not satisfy the minimum burst length requirement can be groomed together. When network load is low, MinRTRR results in the best performance, because Min-
$RTRR$ can groom sub-bursts whose routes may not necessarily overlap completely or partially, and therefore is less restrictive, so that more sub-bursts can be groomed and scheduled at the same time without or with minimum padding waste in this case. With higher offered loads, $NoORW$ outperforms the other two, because the sub-bursts in $MinRTRR$ case may go through a longer route and a groomed burst may need to be transmitted to multiple downstream nodes at the same time, which potentially increases the blocking probability. In 16-node 30-edge network and 40-node 70-edge network, the path lengths between node pairs are more evenly distributed, and more bursts overlap completely or partially in the paths. Therefore, more bursts can be groomed by the grooming algorithms, which potentially decreases the blocking probability. Hence, as for the burst blocking probability performance, the grooming algorithms can perform better in these two network topologies.

Figure 4.4.11: Burst blocking probability under different grooming algorithms for NSF network topology.
Figure 4.4.12: Burst blocking probability under different grooming algorithms for 16-node 30-edge network topology.

C. Average Burst End-to-End Delay

As for the average burst end-to-end delay, the simulation results in Figs. 4.4.14 to 4.4.16 also show that the delay performance with burst grooming is better than that without grooming, because some sub-bursts do not need to be held at the edge nodes until they are timed out when burst grooming is performed. When the grooming algorithms are not applied, all the sub-bursts are assumed to go through the shortest routes, therefore the average burst end-to-end delay remains stable in this case. With low offered loads, MinRTRR can obtain the lowest average burst end-to-end delay. This is because fewest sub-bursts need to be transmitted until they are timed out and the nodes can transmit a groomed burst to multiple downstream nodes simultaneously in this case. When the network load increases, more sub-bursts arrive at the edge nodes, and it is more likely that routes taken by sub-bursts overlap completely or
partially. Hence, NoORW achieves the best performance in this scenario. In general, the average end-to-end delay performance is the best in 16-node 30-edge network, followed by NSF and 40-node 70-edge networks. This is because the end-to-end delay decreases with the decrease of the path lengths and the average path lengths between node pairs in 16-node 30-edge network are shorter than those in the other two networks.

**D. Average Number of Sub-bursts Per Groomed Burst**

Next we look at the performance of the average number of sub-bursts per groomed burst. The simulation results in Figs. 4.4.17 to 4.4.19 demonstrate that in general MinRTRR can aggregate more sub-bursts in a groomed burst than NoORW does. MinRTRR is less restrictive than NoORW in terms of selecting sub-bursts for grooming, which allows more sub-bursts to be groomed in MinRTRR case. It is also shown
Figure 4.4.14: Average burst end-to-end delay under different grooming algorithms for NSF network topology.

that the grooming algorithms can obtain better performance for higher traffic loads. Higher traffic loads generate more sub-bursts, so that sub-bursts are more likely to be groomed together. In 40-node 70-edge network topology, the grooming algorithms can achieve the best performance in terms of average number of sub-bursts per groomed burst. The average path lengths between node pairs in 40-node 70-edge network are longer than those in the other two networks, and more sub-bursts have their paths overlap completely or partially. The path lengths between node pairs are also more evenly distributed in 16-node 30-edge network and 40-node 70-edge network, so bursts have better chances to overlap completely or partially in the paths, which in turn increase the performance of average number of sub-bursts per groomed burst.

**E. Padding Waste**

The performance in terms of padding waste is depicted in Figs. 4.4.20 to 4.4.22.
When the grooming algorithms are applied, the padding waste decreases with the increase of the traffic load. The performance of padding waste is related to the performance specified by average number of sub-bursts per groomed burst. As shown before, the number of sub-bursts in the groomed bursts increases and the size of the groomed burst increases, as the traffic load increases, so the padding waste is less for higher offered loads. In general, the padding waste for MinRTRR case is less than that in NoORW case. The reason is that MinRTRR is less restrictive than NoORW in selecting sub-bursts for grooming, and therefore can groom more sub-bursts than NoORW does. For the non-grooming case, sub-bursts are not groomed before they are transmitted, so the padding waste for non-grooming case is quite stable. The grooming algorithms need the least padding waste in 40-node 70-edge network, which is also shown above that the the grooming algorithms can achieve the best performance in terms of average number of sub-bursts in this network.
Figure 4.4.16: Average burst end-to-end delay under different grooming algorithms for 40-node 70-edge network topology.

**F. Over-Routing Waste**

The performance of over-routing waste is given in Figs. 4.4.23 to 4.4.25. Here, we set the average sub-burst size to 1250 bytes. Note that there exists no over-routing waste in NoORW case or non-grooming case as analyzed before. This performance is also related to the performance specified by the number of sub-bursts per groomed burst, since the over-routing waste may increase if the number of sub-bursts in a groomed bursts increases. When the grooming algorithm MinRTRR is applied, the over-routing waste increases with the increase of the traffic load. The number of sub-bursts in the groomed bursts increases, and more sub-bursts traverse unnecessary links, as the traffic load increases. Therefore, the over-routing waste is higher for heavier traffic loads. Comparing the over-routing waste performance in different network topologies, we can notice that the over-routing waste of MinRTRR case
in 40-node 70-edge network is the most. This observation is in accordance with the simulation results on the performance shown before on the performance of the number of sub-bursts per groomed burst, in which the algorithm $MinRTRR$ achieves the best performance.

4.5 Summary

In this chapter, assuming burst grooming can only be realized at edge nodes, we have studied the burst grooming problem where sub-bursts originating from the same source may be groomed together regardless of their destinations under certain conditions. To achieve more efficient burst grooming, we explore the capability that core nodes can split incoming light signals to support multicast, and deal with the case when sub-bursts in the network is too small to satisfy the minimum burst length requirement. We have proposed two effective burst grooming algorithms, (1) a *no...
Figure 4.4.18: Average number of sub-bursts per groomed burst under different grooming algorithms for 16-node 30-edge network topology.

*over-routing waste* approach (*NoORW*); and (2) a *minimum relative total resource ratio* approach (*MinRTRR*). Our simulation results have shown that the proposed algorithms are effective in terms of the burst blocking probability, the average burst end-to-end delay, the number of sub-bursts per groomed burst, as well as the resource waste.
Figure 4.4.19: Average number of sub-bursts per groomed burst under different grooming algorithms for 40-node 70-edge network topology.

Figure 4.4.20: Padding waste under different grooming algorithms for NSF network topology.
Figure 4.4.21: Padding waste under different grooming algorithms for 16-node 30-edge network topology.

Figure 4.4.22: Padding waste under different grooming algorithms for 40-node 70-edge network topology.
Figure 4.4.23: Over-routing waste under different grooming algorithms for NSF network topology.

Figure 4.4.24: Over-routing waste under different grooming algorithms for 16-node 30-edge network topology.
Figure 4.4.25: Over-routing waste under different grooming algorithms for 40-node 70-edge network topology.
Chapter 5

Physical Impairment Aware Scheduling in Optical Burst Switched Networks

5.1 Introduction

In all-optical networks, an optical signal may traverse a number of intermediate nodes and long fiber segments without any optical-electrical-optical (O/E/O) conversion and signal regeneration, which significantly reduces the overall network cost and increases the transparency. Much previous research on all optical networks assumes an ideal physical optical network. That is, the signal is immune to degradation caused by noise and other impairments. However, during transmission, signal quality is subject to various physical impairments introduced by optical fibers, switching equipment, or other network components [123–125], such as multiplexers/demultiplexers, switches, amplifiers, and so on. Therefore, the noise and signal distortions due to physical impairment effects may accumulate as the signal travels through the network. The transmission performance of optical paths may not always satisfy the service requirements due to signal quality degradation. The degradation may be significant enough
such that the bit-error rate (BER) \cite{126} is unacceptably high at the destination, rendering the signal not usable \cite{123,124}.

While much research has been performed to deal with physical impairment effects at the optical layer (the details of which will be summarized in Section 5.5), physical impairment effects have not received much attention in OBS networks. At the physical layer, physical impairment effects in OBS networks, such as noise, crosstalk of WDM channels, semiconductor optical amplifier (SOA) gain saturation and dynamics, and so on, are investigated in \cite{19}. An analytical model is established in \cite{127} to analyze the burst blocking probability in OBS networks with no wavelength conversion, focusing on polarization mode dispersion (PMD) and amplifier related noise (e.g., amplified spontaneous emission (ASE)). The problem of routing optical signals of various types in an OBS network while maintaining optical signal quality is addressed in the context of Just-In-Time (JIT) signaling protocol in \cite{128}. Another popular signaling protocol in OBS networks is Just-Enough-Time (JET) \cite{20}. To the best of our knowledge, no work has been reported that considers physical impairment effects in scheduling algorithms with the JET signaling protocol. Based on earlier work, we tackle the problem of burst scheduling in OBS networks with JET signaling, taking into account transmission of physical impairments. The objective is to design impairment-aware burst scheduling algorithms. We propose three effective burst scheduling algorithms: (1) a \textit{JET Based Physical Impairment Constrained Algorithm (JETPIC)}, (2) an \textit{Integrated Physical Impairment Constrained Algorithm (IPIC)}, and (3) an \textit{Enhanced Integrated Physical Impairment Constrained Algorithm (EIPIC)}. At an OBS node, the proposed algorithms schedule bursts for transmission
by searching for available resources as well as verifying signal quality. Our simulation results show that the proposed algorithms are effective in terms of reducing the burst blocking probability. Algorithm \textit{JETPIC} outperforms algorithms \textit{IPIC} and \textit{EIPIC} in burst blocking probability with a moderate average end-to-end delay performance. Algorithm \textit{EIPIC} achieves better burst blocking probability performance than algorithm \textit{IPIC} while algorithm \textit{IPIC} attains the lowest end-to-end delay among the three impairment-aware algorithms at the price of a higher burst blocking probability.

The rest of this chapter is organized as follows. Section 5.2 describes the physical impairments considered in this dissertation. Physical impairment aware burst scheduling algorithms are presented in Section 5.3. Section 5.4 reports the performance evaluation. Section 5.5 summarizes the related work. This chapter concludes in Section 5.6.

\section{5.2 Physical Impairment}

Physical impairments in optical networks fall into two classes: linear and nonlinear. Linear impairments, such as fiber dispersion and amplifier noise, are independent of signal power. Nonlinear impairments, such as four-wave mixing (FWM) and cross-phase modulation (XPM), are more complex. Similar to [127, 129, 130], we consider fiber dispersion and amplifier noise in our work because (1) fiber dispersion and amplifier noise are the dominant impairments in a high-speed optical network (with data rate \(\geq 10\ \text{Gb/s}\)); (2) no general analytical model for nonlinear impairments is available [131, 132]; (3) if a burst can not be transmitted with satisfactory quality assurance due to fiber dispersion and amplifier noise, it can not be transmitted by taking
into account extra physical impairment effects; and (4) given more budget (margin) for quality of transmission (QoT), a burst is more likely transmitted successfully by taking more physical impairments into consideration.

5.2.1 Polarization Mode Dispersion

Two most important physical impairments are dispersion and noise. Chromatic dispersion (CD) and polarization mode dispersion (PMD) are the two main sources of dispersion. Chromatic dispersion is a broadening of the input signal as it travels down the length of the fiber. It results from a variation in propagation delay for different wavelengths, and is affected by fiber materials and dimensions. Chromatic dispersion is deterministic in nature, and can be compensated for.

PMD occurs when two different polarizations of light in a waveguide, which normally travel at the same speed, travel at different speeds due to random imperfections and asymmetries, causing random spreading of optical pulses and making it impossible to transmit data reliably at high speeds. Therefore, PMD limits the data rate and the lengths of fibers that signals traverse. Furthermore, PMD is a stochastic and time varying effect, and it is not easy to compensate for in practice. We consider PMD as the main dispersion effect in this work. The *PMD constraint* can be expressed as follows \[ B \times \sqrt{\sum_{k=1}^{H} D_{PMD}(k)^2 \times L(k)} \leq \delta, \] where \( B \) is the data rate, \( D_{PMD}(k) \) is the fiber PMD parameter in the \( k \)-th hop of the signal path, \( H \) is the total number of hops, \( L(k) \) is the fiber length of the \( k \)-th hop, and \( \delta \) is the user requirement parameter which indicates the tolerable limit of

104
the fractional pulse broadening. A burst should be routed on paths under the PMD constraint given by Eq. (5.1).

5.2.2 Amplified Spontaneous Emission

Amplified spontaneous emission (ASE) is produced when a laser gain medium is pumped to produce a population inversion, and simultaneously increasing the optical signal noise level. In optical networks, optical amplifiers are the major noise source on the signal path between two nodes. Only noise generated by ASE is considered in this work. ASE can not be compensated for.

The ASE noise of an amplifier can be approximated by a white, Gaussian random process. The power spectral density with a Gaussian distribution at the amplifier output is [19]:

$$\rho_{ASE} = n_{sp} \cdot (G - 1) \cdot h\nu,$$

(5.2)

where $n_{sp}$ is the spontaneous emission factor, $G$ is the total amplifier gain, $h$ is the Plank’s constant, $\nu$ is the carrier frequency, and $h\nu$ is the photon energy. ASE noise is superposed in both signal polarizations, and the total ASE noise power $P_{ASE}$ is:

$$P_{ASE} = 2 \cdot \rho_{ASE} \cdot B_{tot},$$

(5.3)

where $B_{tot}$ represents the total system bandwidth.

ASE noise can be measured or estimated via optical signal to noise ratio (OSNR). OSNR is defined by

$$OSNR = \frac{P_S}{P_N},$$

(5.4)

where $P_S$ is the signal power and $P_N$ is the noise power. The noise performance of
an amplifier can be characterized by a noise figure $NF$, given by

$$NF = \frac{OSNR_{in}}{OSNR_{out}}, \quad (5.5)$$

where $OSNR_{in}$ and $OSNR_{out}$ are the OSNR at the input and output of the amplifier, respectively. Specifically,

$$NF = \frac{P_{S, in}}{P_{S, out}} \cdot \frac{P_{N, out}}{P_{N, in}} = \frac{1}{G} \cdot \left[1 + \frac{P_{N, in}}{P_{S, in}} \cdot \frac{n_{sp} \cdot h \nu \cdot (G - 1) \cdot B_{tot}}{P_{N, in}}\right], \quad (5.6)$$

where $P_{S, in}$ and $P_{S, out}$ are the signal power at the input and output of the amplifier, respectively, while $P_{N, in}$ and $P_{N, out}$ are the noise power at the input and output of the amplifier, respectively. Moreover,

$$P_{S, out} = P_{S, in} \cdot G \quad (5.7)$$

and

$$P_{N, out} = P_{N, in} + (G - 1) \cdot n_{sp} \cdot h \nu \cdot B_{tot}, \quad (5.8)$$

where $P_{N, in}$ corresponds to the zero point energy $W_0$ of the quantum mechanic oscillation or vacuum fluctuations, and is defined as

$$P_{N, in} = W_0 = \frac{1}{2} \cdot h \nu \cdot B_{tot}. \quad (5.9)$$

Therefore, the noise figure of an amplifier can be defined as

$$NF = \frac{1}{G} \cdot \left[1 + 2 n_{sp} \cdot (G - 1)\right] \quad (5.10)$$

or

$$NF \approx 2 n_{sp} \quad (5.11)$$
for $G \gg 1$.

The $NF$ of a signal path consisting of $M$ consecutive amplifiers can be calculated as

$$NF_p = NF_1 + \frac{NF_2 - 1}{G_1} + \frac{NF_3 - 1}{G_1 \cdot G_2} + \cdots + \frac{NF_M - 1}{G_1 \cdot G_2 \cdots G_{M-1}},$$

(5.12)

where $NF_i (1 \leq i \leq M)$ and $G_i (1 \leq i \leq M, G_i \gg 1)$ are the noise figure and amplifier gain of the $i$-th amplifier, respectively [9, 19].

We can calculate the noise figure of a signal path using Eq. (5.12), and then decide whether the noise level is qualified by comparing it with a prespecified noise threshold.

### 5.2.3 Bit-Error Rate

Bit-error rate (BER) is a primary metric for quantifying signal quality, and it is measured by the ratio of the number of erroneous bits detected to the number of transmitted bits. Q-factor can be used as an intermediate estimator for BER and OSNR [9, 131]. The relation between Q-factor and OSNR can be approximated by

$$Q \approx \sqrt{\frac{B_o}{B_e \sqrt{4OSNR + 1}}} \frac{2OSNR}{1 + 1},$$

(5.13)

where $B_o$ and $B_e$ are the optical and electrical bandwidth, respectively [130]. The relation between Q-factor and BER is given by

$$BER = \frac{1}{2} erfc \left( \frac{Q}{\sqrt{2}} \right) \approx \frac{1}{Q \sqrt{2\pi}} \exp \left( -\frac{Q^2}{2} \right),$$

(5.14)

where $erfc$ is the complementary error function [19].

In fact, OSNR is directly correlated to BER via the following equation:

$$BER(OSNR) = \frac{1}{\sqrt{2\pi}} \int_{OSNR}^{\infty} \exp \left( -\frac{t^2}{2} \right) dt$$

(5.15)
A lower OSNR means a higher BER, i.e., a worse signal quality.

5.3 Physical Impairment Aware Scheduling Algorithms

Realistically, the routing and transmission of a burst in an OBS network should take into account physical impairment constraints. At a node, a burst may be blocked due to two reasons: no free resource and quality of transmission (QoT) constraint. In the first case, a burst may be blocked when no wavelength on the route is available to accommodate the burst. This case is termed as resource blocking. In the second case, a burst has to be dropped if the signal quality cannot satisfy the requirement due to physical impairment effects, even if there may be resources available. This second case is termed as QoT blocking. In this chapter, we consider PMD and ASE impairment effects, and no deflection routing using FDLs is used.

With network physical impairment awareness, a burst should be dropped when the signal quality has fallen below the requirement at an intermediate node so that the resource waste due to unnecessary burst transmission can be avoided. Hence the burst blocking probability can potentially be reduced.

Assume that the network under consideration disseminates network state information (e.g., via a link state routing protocol) so that nodes in the network learn additional network state information, such as the length and PMD parameters of each link, the number of amplifiers on each link, the gain and spontaneous emission factor of each amplifier, and so on. Upon the arrival of a burst control packet, an OBS node runs a physical impairment aware burst scheduling algorithm to schedule
A burst may associate with it a PMD path-constraint parameter \( \delta \) (Eq. (5.1)). An ingress node can also determine the minimum OSNR requirement of a signal, \( OSNR_{\text{min}} \), based on the BER requirement of a client. The OSNR of the burst-carrying signal should not fall below \( OSNR_{\text{min}} \) during the transmission of the bursts of the client, that is

\[
OSNR \geq OSNR_{\text{min}}.
\]  

(5.16)

We propose three physical impairment aware burst scheduling algorithms using the JET signaling protocol: (1) JET based physical impairment constrained algorithm (\textit{JETPIC}), (2) integrated physical impairment constrained algorithm (\textit{IPIC}), and (3) enhanced integrated physical impairment constrained algorithm (\textit{EIPIC}). The general burst scheduling procedure is shown in Fig. 5.3.1. The verification of signal quality is performed when control packets (CPs) are processed. When a CP is received, an OBS node routes the data burst with one of the algorithms. These algorithms incorporate the physical impairment constraints in the context of the JET signaling protocol, and then they are used in combination with any existing burst scheduling algorithms, such as FF, LAUC-VF, and so on.

5.3.1 JET Based Physical Impairment Constrained Algorithm

The JET based physical impairment constrained algorithm (\textit{JETPIC}), shown in Fig. 5.3.2, extends existing scheduling algorithms under the JET signaling proto-
Figure 5.3.1: Physical impairment aware burst scheduling process in an OBS network.

col to take into account the physical impairment constraints. This algorithm has two steps: (1) free resource search, and (2) QoT verification. Upon the arrival of a burst control packet at node $v$, node $v$ tries to find free local resources to schedule the incoming burst. In the free resource search step, the scheduling algorithm can adopt any existing impairment-unaware scheduling algorithm, such as FF, LAUC-VF, and so on. If no free resource is available to accommodate the burst, the burst is dropped due to resource blocking. If free resources are successfully found, the end-to-end signal quality is further estimated. The QoT verification checks whether the PMD and OSNR constraints along the path from the source to the destination through node $v$ (with a shortest path from $v$ to the destination) are satisfied. The signal quality is unacceptable if at least one of these two constraints cannot be satisfied.

The PMD constraint is estimated via Eq. (5.1) using link state information, i.e., PMD parameters and link length information. If the end-to-end PMD constraint
cannot be satisfied, QoT verification fails; otherwise, the end-to-end OSNR constraint is tested via Eq. (5.16) based on the gain and ASE factor parameters of amplifiers along the burst transmission path. If the OSNR of the incoming burst is smaller than the prespecified $OSNR_{\text{min}}$, the signal quality fails the QoT verification; otherwise, the resources determined in the first step are verified to be acceptable in terms of signal quality.

If the resources determined in the first step do not satisfy the physical impairment constraints, they will be marked as unavailable for this burst, and node $v$ repeats the two-step process of free resource search and QoT verification as stated above using another shortest path from node $v$ to the destination. A maximum number of iterations of the above process is imposed because of the scheduling time constraint. If the number of iteration has exceeded the maximum allowed and no satisfactory path in terms of signal quality with free resources can be found, the burst is dropped. On
the other hand, if the available resources satisfy the physical impairment constraints, the burst is scheduled at node \( v \), and the control packet is updated and forwarded.

Therefore, a control packet needs be augmented to carry the cumulative PMD and OSNR along the path traversed. Specifically,

\[
\sum_{k=1}^{H'} D_{PMD}(k) \times L(k) \quad \text{and} \quad NF_1 + \frac{NF_2 - 1}{G_1} + \frac{NF_3 - 1}{G_1 \cdot G_2} + \cdots + \frac{NF_{M'} - 1}{G_1 \cdot G_2 \cdot \cdots \cdot G_{M'-1}}
\]

are included in the control packets, where \( H' \) and \( M' \) are the number of links and the number of amplifiers that the burst has traversed, respectively. When a burst can be scheduled subject to the physical impairment constraints, these two cumulative values are updated before the control packet gets forwarded.

**Time Complexity** Algorithm JETPIC running at node \( v \) may attempt \( k \) shortest paths from \( v \) to the destination in order to discover a QoT qualified path with locally available resources. The time complexity of the Dijkstra algorithm is \( O(|V|^2 + |E|) \) where \( |V| \) is the number of nodes and \( |E| \) is the number of links in the network. Assume that burst scheduling algorithm LAUC-VF is used which has a time complexity of \( O(W \log N_b) \) where \( W \) is the number of wavelengths on a link, and \( N_b \) is the maximum number of bursts scheduled on a wavelength. Further assume that a burst travels a maximum of \( N_H \) hops in the network, and the maximum number of optical amplifiers along the transmission path of a burst is \( N_a \). The time complexity of signal quality verification process is \( O(N_h + N_a) \) operations. Therefore, the worst case time complexity for algorithm JETPIC is \( O(k|V|^2 + |E| + W \log N_b + N_h + N_a) \) where \( k \) is the number of shortest paths searched.
5.3.2 Integrated Physical Impairment Constrained Algorithm

The previous algorithm JETPIC performs two steps. A burst succeeds in the first step of free resource search may be dropped in the second step of signal quality verification due to unqualified PMD constraint and/or OSNR constraint. Note that OSNR constraint verification is not necessary if PMD constraint cannot be satisfied. Therefore, we can improve algorithm JETPIC by taking into account the PMD effect in the path computation. That is, we can integrate the PMD effect into the link cost. Specifically, the cost of a link \( e \) is assigned a value of \( D_{PM}^2(e) \cdot L(e) \) where \( D_{PM}(e) \) and \( L(e) \) are the link PMD parameter and length, respectively. The flow diagram of the improved algorithm, integrated physical impairment constrained algorithm (IPIC), is shown in Fig. 5.3.3.

**Algorithm 1 CreateAuxGraph(G, CP)**

**Require:** Network topology, \( G = (V, E) \); control packet of incoming burst, CP

**Ensure:** Auxiliary graph, \( G' = (V', E') \)

1. \( G' = G \);
2. In \( G' \), remove the predecessor node on the path traversed by the CP and links connected to this node, resulting a new \( G' \);
3. **for all** \( e \in E' \) **do**
   4. Update the cost of \( e \) as \( D_{PM}^2(e) \cdot L(e) \) in \( G' \);
5. **end for**
6. Return \( G' \);

When a burst control packet arrives at node \( v \), an auxiliary graph \( G' \) is created, as illustrated in Algorithm 1, to help determine a shortest path from node \( v \) to the destination. Assume that the network topology is \( G = (V, E) \), where \( V \) the network node set, and \( E \) is the link set. To avoid local loops, the auxiliary graph \( G' = (V', E') \) is constructed by removing the predecessor node on the path traversed by the CP and
Figure 5.3.3: Flow diagram of the integrated physical impairment constrained algorithm (IPIC).

links connected to this node. In addition, for each link $e \in E'$ with length $L(e)$ and link PMD parameter $D_{PMD}(e)$, the cost of this link is assigned as $D_{PMD}^2(e) \cdot L(e)$ in $G'$.

The shortest path from node $v$ to the destination in the auxiliary graph $G'$ may not satisfy the QoT requirement. Thus, the path is then tested for signal quality. If the path cannot satisfy the PMD constraint, the burst is dropped because no other path can satisfy the PMD constraint.

If the path satisfies the PMD constraint, the OSNR of the path is evaluated in the same way as in algorithm JETPIC. If the path cannot satisfy the OSNR constraint, it will be marked as signal quality unacceptable. In this case, the node will try to find another shortest path, and the QoT constraints will also be evaluated. If the
signal quality is satisfactory, the algorithm searches for free local resources for burst transmission to the next hop. If no free resource is available, the path is marked as resource unavailable. In this case, the algorithm looks for another shortest path, and repeats the path computation, QoT verification, and free resource search process. Similar to algorithm JETPIC, there also should be a maximum iterations of this process.

Once the burst is scheduled, the control packet is updated in the same way as that in JETPIC. Algorithm IPIC is designed to integrate the testing of the PMD constraint with path computation.

**Time Complexity** The time complexity of link cost assignment procedure in the auxiliary graph creation process is $O(|E|)$. Hence, algorithm IPIC has a time complexity of $O(|E|) + O(k|V|^2 + |E| + W \log N_h + N_h + N_a) \approx O(k|V|^2 + |E| + W \log N_b + N_h + N_a)$.

### 5.3.3 Enhanced Integrated Physical Impairment Constrained Algorithm

The previous two algorithms may have to *dynamically* find multiple alternative paths with a path searching algorithm for every burst. For example, in algorithm IPIC, for each burst, the algorithm creates an auxiliary graph, and searches for the paths one by one if necessary, which can be time consuming.

At a node, $k$-shortest paths from this node to all the other network nodes can be computed and stored in advance. On the arrival of a burst control packet, the network node can then try to schedule the incoming burst on one of the stored shortest paths as shown in Fig. 5.3.4. This is more time efficient because there is no need to perform
Figure 5.3.4: Flow diagram of the enhanced integrated physical impairment constrained algorithm (EIPIC).

path computation for each burst. The node only needs to retrieve the stored paths to check whether the signal quality is satisfied and free local resources are available. Specifically, after receiving link state updates periodically, a network node may compute $k$-shortest paths to all the other edge nodes, with the cost of link $e$ assigned as $D^2_{PMD}(e) \cdot L(e)$. Upon the arrival of a burst control packet, the node retrieves the first shortest path and checks whether the end-to-end path is QoT-qualified and free local resources are available on the retrieved path for burst transmission, in a way similar to that of algorithm IPIC. If the burst cannot be scheduled on this path, other shortest paths may be tried in the ascending order of the weighted path length. If the PMD constraint cannot be satisfied, the burst is dropped without trying other alternative paths because no other path is qualified in terms of the PMD constraint. In the case that the burst cannot be scheduled on any of these $k$-shortest paths, the
burst is dropped.

**Time Complexity** Since the $k$-shortest path computation is performed at the network initialization or periodically, its impact on the time complexity of burst scheduling is minimal. Therefore, the time complexity of algorithm EIPIC is approximately $O(k[W \log N_b + N_h + N_a])$.

### 5.4 Performance Evaluation

#### 5.4.1 Simulation Setup

We evaluate the performance of our proposed algorithms (JETPIC, IPIC, and EIPIC) by implementing the algorithms using ns-obs version 0.9a [122]. The network topologies used in the simulation are the NSF network topology (Fig. 5.4.5), 16-node Torus network topology (Fig. 5.4.6), and 8-node ring network topology (Fig. 5.4.7) with link length in kilometers. In these two networks, each link is bi-directional with a fiber in each direction. The number of data wavelengths and control wavelengths on each link are 8 and 2, respectively. The bandwidth of each wavelength is 10 Gb/s. An amplifier is applied every 100 km. The optical fibers can transmit light at about the speed of 200,000 km/s. Self-similar traffic is generated and is uniformly distributed between all pairs of edge nodes.

The parameters used in the simulations are as follows:

- Amplifier gain: 15 dB;
- ASE factor $n_{sp}$: 1.5;
- $D_{PMD}(k)$: 0.2 $ps/(km)^{1/2}$;
• Fractional pulse broadening parameter $\delta$: 0.1;

• $OSNR_{\text{min}}$: 7.4 dB ($\text{BER} = 10^{-9}$);

• Offset time: 20 ms;

• Number of shortest paths $k$ to be searched during burst scheduling: 3.

In Torus and ring network topologies, link PMD parameter $D_{\text{PMD}}(k)$ of the links with length 1000 km is $0.1 \text{ ps/(km)}^2$. In ring network topology, the number of shortest paths $k$ to be searched during burst scheduling is 2.

The performance is measured by two metrics: (1) burst blocking probability, and (2) average burst end-to-end delay. The simulation is run with different offered loads which are defined by the following equation [133]:

$$\rho = \frac{N_{IE} \cdot h \cdot r}{C \cdot (2L)} \quad (5.17)$$

where $N_{IE}$ is the number of ingress-egress node pairs, $h$ is the average number of hops of the shortest paths, $r$ is the incoming data rate, $C$ is the link transmission rate, and $L$ is the number of bi-directional links in the network.
Figure 5.4.6: The 16-node Torus network topology.

In the simulation, the burst scheduling on a given link is performed by the latest available unscheduled channel with void filling (LAUC-VF) algorithm [53]. We compare the three impairment-aware algorithms with the impairment-unaware algorithm (LAUC-VF). Bursts are scheduled without deflection routing for the impairment-unaware scheduling algorithm. At the destination, the signal quality is checked, and the burst is dropped if the signal quality is not satisfied.

### 5.4.2 Burst Blocking Probability

The burst blocking probability performance versus different offered loads is depicted in Figs. 5.4.8 to 5.4.10. The simulation results show that the impairment-aware algorithms result in overall significantly reduced burst blocking probabilities than the impairment-unaware algorithm because the impairment-aware algorithms take into account physical impairment effects, and only bursts with good quality transmission
paths are scheduled. That is, unnecessary transmission of QoT-unqualified bursts can be avoided, which saves resources for transmitting bursts whose QoT constraints can be met. Furthermore, the impairment-aware scheduling algorithms search multiple alternative paths during scheduling to accommodate a burst, therefore reducing the burst blocking probability.

In NSF network topology, these three impairment-aware scheduling algorithms achieve the same performance. PMD parameters are the same for all the links, so the alternative paths found by these three impairment-aware scheduling algorithms are the same. Accordingly, these three impairment-aware scheduling algorithms have the same blocking performance.

On the other hand, in Torus and ring network topologies, the longer links have smaller link PMD parameters, while the shorter links have bigger link PMD parameters. Therefore, the shortest paths found by algorithm JETPIC are different from
Figure 5.4.8: Burst blocking probability under different scheduling algorithms for NSF network topology.

those found by algorithms IPIC and EIPIC. Note that the simulation results of algorithms IPIC and EIPIC are the same as illustrated in Fig. 5.4.8 and Fig. 5.4.9. These two algorithms differ in that algorithm EIPIC moves the auxiliary graph creation and multiple shortest paths computation procedure forward to allow path pre-computation. Therefore, as expected, the blocking probability performance for IPIC and EIPIC is the same, and the blocking probability performance curves for these two algorithms overlap with each other.

The incorporation of PMD effects into link cost makes algorithms IPIC and EIPIC more likely to find better outgoing links in terms of signal transmission quality than algorithm JETPIC at an intermediate node. Therefore, algorithms IPIC or EIPIC achieve the best blocking performance at low offered loads in Torus network topology. However, bursts are more likely scheduled on those links with good transmission quality using algorithms IPIC or EIPIC, potentially resulting in congestion on those
In the meantime, algorithm JETPIC tries to schedule the bursts with moderate multiple alternative paths search capability, which potentially balances the load on the network. Hence, algorithm JETPIC leads to better burst blocking performance than algorithms IPIC and EIPIC, as the offered load increases in Torus and ring network topologies.

### 5.4.3 Average Burst End-to-End Delay

The simulation results for average burst end-to-end delay are depicted in Figs. 5.4.11 to 5.4.13. For the impairment-unaware algorithm, generally the average burst end-to-end delay decreases as the network offered load increases. Bursts which need to traverse more hops in the network have a higher probability of being dropped. Consequently, bursts that traverse more hops constitute a smaller portion in the total number of bursts successfully received by the destinations at a higher offered load.
Figure 5.4.10: Burst blocking probability under different scheduling algorithms for ring network topology.

load because more bursts compete for the limited resources, and the bursts with more hops are more likely subject to burst drop.

Similar to burst blocking probability performance, the average end-to-end delay for the impairment-aware algorithms is the same in NSF network topology, because they have the same free resource search and signal quality verification behavior. In Torus network topology, the average end-to-end delay for algorithm JETPIC increases as the offered load increases. This is because bursts are more likely deflected onto longer-distance links with smaller link PMD parameters where there is free resource and QoT constraints can be satisfied. As stated above, algorithms EIPIC and IPIC are more aggressive than algorithm JETPIC in finding QoT-qualified free resources for bursts, and bursts are more likely to go through longer path with lower impairment impact, so that the average end-to-end delay for these two algorithms are higher than that for algorithm JETPIC. Similarly, in ring network topology, algorithms IPIC and
Figure 5.4.11: Average burst end-to-end delay under different scheduling algorithms for NSF network topology.

EIPIC result in a larger average end-to-end delay than algorithm JETPIC because algorithms IPIC and EIPIC prefer paths that are longer but have lower physical impairment effect.

5.5 Related Work

Much research has been performed to deal with the physical impairment effects at the optical layer. Optical signal-to-noise ratio (OSNR) optimization problem is formulated as an OSNR Nash game for each link to adjust the link-level power in [134, 135]. A method is proposed in [136] to suppress the in-band amplified spontaneous emission (ASE) noise from optical amplifiers using a spectral spread technique. If the signal spectrum is spread by modulation, half of the in-band ASE merged into the spectral-spread signal is eliminated through the reverse modulation process. Polarization mode dispersion (PMD) mitigation of 8.6ps mean differential group delay (DGD)
Figure 5.4.12: Average burst end-to-end delay under different scheduling algorithms for Torus network topology.

at 43 Gbps differential phase shift keying (DPSK) is demonstrated by a straight-line cascade of 5 distributed polarization scramblers and enhanced FEC (UFEC) in [137]. PMD mitigation is studied for all channels in a WDM system with FEC using distributed fast polarization scrambling in [138].

There has been some research that considers the physical impairments in optical circuit switching networks. Crosstalk (XT) and ASE are the main physical impairments considered when the BER of a candidate path is evaluated in [123]. The work of [139] and [140] deals with PMD when considering the quality of the optical links. A centralized approach, two distributed approaches, and a hybrid approach that integrate information about most relevant physical impairments, PMD and OSNR, in routing and wavelength assignment and lightpath provisioning are presented and assessed in [141]. Multiple physical impairments are checked in [142], such as amplifiers, switch equipment, and multiplexer/demultiplexer characteristics.
Physical impairment constrained routing and wavelength assignment algorithms have been proposed to choose feasible paths in terms of signal quality while minimizing the blocking probability. A hierarchical routing and wavelength assignment model is investigated in [142] for high-speed connection provisioning where the OSNR and PMD effects are estimated at the physical layer. In [143], based on the integrated consideration of signal transmission impairments and service classification, a multi-path routing and wavelength assignment algorithm is adopted to set up an appropriate lightpath that matches the request priority. By analyzing the transmission quality of lightpath candidates, differentiated QoS in the optical domain is provided.

Two impairment-aware routing and wavelength assignment algorithms are proposed in [129, 130] that consider OSNR requirement and PMD effect as signal-quality constraints. Distributed connection provisioning schemes with BER and delay considerations are compared in [144] using two BER estimation procedures and several
wavelength assignment algorithms. A set of routing and wavelength assignment algorithms that mitigate the crosstalk effects are provided in [145]. A reinforcement learning technique is proposed in [146] to choose a tentative lightpath among a set of alternates based only on the past events seen locally in a distributed fashion with physical impairment consideration of ASE, crosstalk, and inter-symbol interference.

At the physical layer, physical impairment effects on OBS networks, such as noise, crosstalk of WDM channels, semiconductor optical amplifier (SOA) gain saturation and dynamics, and so on, are investigated in [19]. An analytical model is established in [127] to analyze the burst blocking probability in OBS networks with no wavelength conversion, focusing on PMD and amplifier related noise (e.g., amplified spontaneous emission (ASE)). The problem of routing optical signals of various types in an OBS network while maintaining optical signal quality is addressed in the context of Just-In-Time (JIT) signaling protocol in [128]. Another popular signaling protocol in OBS networks is Just-Enough-Time (JET) [20]. This chapter is the first to tackle the problem of burst scheduling in OBS networks with the JET signaling protocol, taking into account physical impairments in signal transmission.

5.6 Summary

Optical signal transmission in OBS networks is subject to various physical impairments introduced by the optical fibers, switching equipment, or other network components. The signal degradation due to physical impairment effects may be so significant that the bit-error rate is not acceptable at the destination, rendering the received signal not usable. In this chapter, taking into account physical impairment effects, we
have studied the problem of burst scheduling in OBS networks. We have proposed three effective burst scheduling algorithms, (1) a \textit{JET Based Physical Impairment Constrained Algorithm (JETPIC)}, (2) an \textit{Integrated Physical Impairment Constrained Algorithm (IPIC)}, and (3) an \textit{Enhanced Integrated Physical Impairment Constrained Algorithm (EIPIC)}. Our simulation results show that the proposed algorithms are effective in terms of reducing the burst blocking probability. In general, algorithm \textit{JETPIC} outperforms algorithms \textit{IPIC} and \textit{EIPIC} in burst blocking probability and average end-to-end delay performance.
Chapter 6

Physical Impairment Aware Scheduling in Dual Control Packets Optical Burst Switched Networks

6.1 Dual-Header Optical Burst Switching

Each control packet in OBS networks contains information of the corresponding burst, such as burst arrival time, burst length, incoming wavelength, and so on. Control packets are processed electronically ahead of burst arrival at core nodes so that resources are reserved for the incoming bursts and switches are configured before bursts arrive. Offset time is the time between burst scheduling and burst arrival. Control packets incur processing delay when they traverse each node in OBS networks, and the offset time of a burst shrinks along the transmission path. Therefore, the offset time of bursts varies in traditional OBS networks that employ one control packet for each burst. This type of OBS networks is called single header OBS networks. Variable offset time may result in bursts not being serviced in first-come-first-serve (FCFS) order in OBS networks. The out-of-order scheduling leads to voids or fragmentation on the outgoing wavelength channels, which can severely degrade the network
throughput and blocking probability performance, if not dealt with carefully.

Scheduling algorithms have been proposed to improve the network performance [53, 92, 147], such as FF-VF, LAUC-VF, Min-SV, and so on. In order to cope with variable offset time and voids on wavelength channels, these algorithms take more time to schedule an incoming burst. These algorithms can decrease the blocking probability and improve the network throughput. However, the performance improvement is at the price of higher scheduling complexity.

It turns out that the out-of-order scheduling of burst and voids/fragments on wavelength channels are artifacts of variable burst offset time. Many of the scheduling complexity and performance issues can be avoided in OBS networks if OBS core can control burst offset time. For example, FDLs can be employed to buffer bursts in order to compensate for the control header processing time. However, the physical properties of FDLs prevent FDLs from providing continuous delay time.

A signalling architecture called Dual-header Optical Burst Switching (DOBS) is proposed in [148, 149] to reduce the complexity of scheduling algorithms. In DOBS networks, two control packets are used for each burst, which decouples the resource request from resource reservation as shown in Fig. 6.1.1. Due to this decoupling, each node can specify the time for burst scheduling operations and individually select the offset time termed as the functional offset time of each burst without FDLs. By selecting the same functional offset time for each burst at each node, the variable offset time can be avoided, and bursts can be serviced in FCFS order, which makes simple burst scheduling possible.
Figure 6.1.1: Illustration of the dual-header OBS signalling.

### 6.1.1 Control Information Division

In DOBS networks, two control packets are used for each data burst. The first control packet \( CP_1 \) contains information about the service requirements of the burst, e.g., the routing and temporal information of the data burst. The second control packet \( CP_2 \) contains the physical information of the incoming burst, e.g., the incoming wavelength index, which is used to configure the optical switch at a core node.

At time \( t_{REQ} \), the first control packet for a data burst arrives at a node, and it is processed immediately and forwarded to the downstream node. The processing of \( CP_1 \) includes determining the outgoing link, updating the temporal information of the data burst, and so on.

The node performs burst scheduling at time \( t_S \) to determine the specific outgoing wavelength. Therefore, the outgoing wavelength selection is independent from the incoming wavelength. If the node finds a wavelength to accommodate the incoming data burst, the second control packet is transmitted to the downstream node to
advertise the incoming wavelength used by the burst. At time $t_{RSV}$, $CP_2$ is received, and the node can configure the switch according to the incoming wavelength and the selected outgoing wavelength during the burst scheduling process.

### 6.1.2 Signalling in DOBS Networks

![Diagram showing end-to-end signalling in DOBS networks.]

Figure 6.1.2: The end-to-end signalling in DOBS networks.

The end-to-end signalling for a burst that traverses 3 hops is depicted in Fig. 6.1.2. After the first control packet arrives at a node, it is processed immediately and forwarded to the downstream node. The information about the data burst is stored at this node until burst scheduling time. After the burst is scheduled successfully, the second control packet is transmitted to the downstream node so that the downstream node can configure the switch. In general, the physical offset time, which is defined as the time between the first control packet and the arrival of the burst, shrinks as the first control packet traverses its path, while the functional offset can be selected independently.
6.1.3 Burst Scheduling

If the functional offset time for each burst is the same at an outgoing link, bursts can be serviced in FCFS order. In such a system, all best-effort, non-preemptive wavelength selection algorithms will result in identical blocking performance. A simple scheduling algorithm, *free-channel queue* (FCQ), can be implemented. The time complexity of this algorithm is $O(1)$.

Each node stores a list of all the channels that are available for burst reservation in a *free channel queue*. At a burst scheduling time $t_S$, if FCQ is empty, no channel is available for this data burst. If the queue is not empty, the burst is scheduled on the channel at the head of the FCQ at time $t_S$. That channel is removed from the FCQ and put back into the FCQ at time $t_S + l_b$, where $l_b$ is the burst duration. Thus, the scheduling operation requires only $O(1)$ time.

6.2 Impairment-Aware Burst Scheduling in DOBS Networks

In general, an impairment-aware burst scheduling algorithm is more complicated than a scheduling algorithm counterpart that does not consider physical impairments. The impairment-aware scheduling algorithm not only needs to search for free resources to accommodate the incoming burst, it also needs to check whether the signal quality is qualified when using the resources. A measure or estimation of the signal quality must be performed for each burst or path. A shortest transmission path may not satisfy the physical impairment constraints so that alternative paths may be needed. Impairment-aware scheduling algorithms may also have higher time complexity.
Note however that high-speed OBS networks demand an efficient burst scheduling algorithm as the ultra-high data rate leaves little time for burst scheduling and resource reservation. In the traditional single header OBS networks, a good scheduling algorithm needs to strike a proper balance between network performance and scheduling complexity due to the variable burst offset time and out-of-order scheduling.

We note that burst scheduling is simple and fast in DOBS networks. However in DOBS networks, $CP_1$ may be forwarded to the downstream node without waiting for the result of burst scheduling. Therefore, a burst may be successfully scheduled at a downstream node, while it is blocked in an upstream node. This incurs so called *phantom burst resource waste*.

In this chapter, we take advantage of the DOBS signaling to design a simple impairment-aware OBS scheduling algorithm, and at the same time, perform admission control during the first control packet processing to reduce phantom burst resource waste and achieve good burst blocking performance.

The rest of this chapter is organized as follows. Physical impairment-aware burst scheduling algorithm is presented in Section 6.3. Section 6.4 reports the performance evaluation. This chapter concludes in Section 6.5.

### 6.3 Physical Impairment Aware Scheduling Algorithms

**Definition 6.1** Given two bursts $a$ and $b$ with their burst durations expressed by service time intervals $[t^a_s, t^a_e)$ and $[t^b_s, t^b_e)$, respectively, if there exists time $t$, such that $t \in [t^a_s, t^a_e)$ and $t \in [t^b_s, t^b_e)$, we say bursts $a$ and $b$ conflict with each other; otherwise,
we say bursts $a$ and $b$ are compatible with each other.

Assume bursts use precomputed paths. A burst first attempts to reach its destination along a primary path, which has the minimum path cost. However, if the burst conflicts with another burst or QoT requirement can not be satisfied, the burst may be deflected using an alternate route.

We consider the same physical impairments as presented in Section 5.2. As stated in Section 5.3, a burst may associate with it a PMD path-constraint parameter $\delta$ (Eq. (5.1)). An ingress node can also determine the minimum OSNR requirement of a signal, $OSNR_{\text{min}}$, based on the BER requirement of a client. The OSNR of the burst-carrying signal should satisfy $OSNR$ constraint (Eq. (5.16)).

The proposed impairment-aware scheduling algorithm consists of three parts: (1) offline primary route computation, (2) offline deflection route computation, and (3) online burst scheduling. The first part is to compute the optimal path in terms of path cost for each possible source-destination pair. When conflict occurs, the burst may be deflected using a deflection route which is sub-optimal in terms of path cost. Upon arrival of control packets, online scheduling is to find free resources that qualify for the QoT requirements to transmit the incoming bursts.

### 6.3.1 Primary Route Computation

We model the network topology as a directed graph $G = (V, E)$, where $V$ is the network node set and $E$ is the link set. Each link $e \in E$ is associated with a cost.

The minimum cost path to each destination can be computed, and next hop node
information for each source-destination pair is stored in the primary routing table at each node along the primary path. The shortest paths from a source node to all the destinations can be computed by the Dijkstra algorithm. The time complexity of Dijkstra algorithm is \( O(|V|^2 + |E|) \), where \(|V|\) is the number of nodes and \(|E|\) is the number of links in the network.

### 6.3.2 Deflection Route Computation

**Definition 6.2** Path-degree of a link \( e \in E \), \( \deg(e) \), is the number of primary paths that traverse link \( e \) for all source-destination pairs.

During computation of deflection routes, we can integrate the PMD effect into the link cost. Furthermore, in order to balance the traffic in the network, we take path-degree of each link into account when computing the deflection route for a burst destined to node \( d \), if the primary output link \( e \) at a node \( v \) is not usable for a burst either due to resource blocking or QoT blocking. Specifically, we create an auxiliary graph \( G^* \) for each link \( e \), as illustrated in Algorithm 2. The Dijkstra algorithm can be applied to this auxiliary graph to compute the shortest path from node \( v \) to node \( d \) to obtain the next hop information, which is stored in the deflection routing table.

### 6.3.3 Burst Scheduling

In DOBS networks, the first control packet \( CP_1 \) contains information about the service requirement of a burst, such as burst arrival time, burst length, QoT requirement parameters which is added to implement our impairment-aware scheduling, and so
Algorithm 2 CreateAuxGraph($G^*$, $e$)

Require: the network topology $G = (V, E)$, and link $e$
Ensure: the created auxiliary graph $G^* = (V^*, E^*)$

1: $G^* = G$
2: Remove link $e$ from $G^*$
3: for all $e^* \in E^*$ do
4: Update the cost of $e^*$ as $D_{PMD}^2(e^*) \cdot L(e^*) \cdot \deg(e^*)$ in $G^*$
5: end for
6: Return $G^*$

on; while the second control packet $CP_2$ contains the burst physical information, i.e., incoming wavelength, which is used to configure the optical switch.

Upon the arrival of a $CP_1$, after looking up the primary routing table to find the next hop, the core node first performs admission control that estimates whether free resources are available and the QoT requirements are satisfied. If the primary output link is not feasible due to either resource blocking or QoT blocking, the core node retrieves the deflection route in the deflection routing table and performs QoT verification. If free resources that satisfy the QoT requirements are available, $CP_1$ is sent to the downstream node after updating the carried information. $CP_2$ is transmitted to the next hop after the outgoing wavelength channel is determined by the burst scheduling process.

In DOBS networks, the first control packet for a burst is forwarded to the downstream node before scheduling is performed. During scheduling, each node schedules the burst independently without the knowledge of whether the burst is scheduled successfully or not at the upstream and downstream nodes. Therefore, a burst may be able to obtain available resources at a downstream node while it is blocked due to resource blocking at an upstream node, which incurs so called phantom burst resource
Example 4 In Fig. 6.3.3, the first control packet $CP_1$ for data burst $DB$ is forwarded from node 1 to node 2 before the result of scheduling is obtained at node 1. When node 2 schedules the burst, node 1 fails in scheduling the burst while node 2 successfully allocates resources for $DB$. Obviously, resource waste is incurred by resource allocation for $DB$ at node 2 because $DB$ will be dropped at node 1 and it can not reach node 2.

Therefore, admission control is necessary to guarantee that resources can be successfully allocated for a burst before the first control packet for this burst is passed on to the downstream node. Two components are needed to achieve burst scheduling: (1) admission control and (2) resource allocation as shown in Fig. 6.3.4.

Upon the arrival of the first control packet for a data burst, a network node $v \in V$ first performs admission control. If a burst is admitted, node $v$ forwards the first control packet to the downstream node. At the functional offset time before the arrival of the data burst, node $v$ selects the outgoing wavelength for the incoming burst and transmits the second control packet to the downstream node.
Figure 6.3.4: Components of burst scheduling.

**Admission Control**

In OBS networks, each burst $DB$ has a start time $t_{sDB}$ and an end time $t_{eDB}$, we can express the burst duration using an interval $[t_{sDB}, t_{eDB})$.

Two or more bursts can not be scheduled on the same wavelength on the same link at the same time, if they conflict with each other. When a conflict occurs, only one of the bursts involved can be scheduled.

Therefore, a new burst can be admitted only if the admission of the new burst does not exceed the physical constraint of the link.

Assume that the number of wavelengths on each link is $W$. At any time, at most $W$ bursts can be scheduled on a given link. When the first control packet $CP_1$ for a data burst $DB$ arrives at a node $v$, node $v$ performs admission control by checking whether the admission of burst $DB$ exceed the physical constraint of the link during
the service time of burst $DB$. We divide scheduling time into *time slots*, each of which has the same time span $\Delta$. A time slot $i$ corresponds to the time interval between $(i-1)\Delta$ and $i\Delta$. $CP_1$ is placed in one or more time slots according to the arrival time and burst duration of the incoming burst. For example, in Fig. 6.3.5, $DB_0$ is placed in time slots 0 and 1, while $DB_1$ occupies 3 time slots from slot 1 to slot 3. Note that two bursts that are placed in the same time slot may not conflict with each other, so that they can be allocated the same wavelength, such as bursts $DB_0$ and $DB_4$ in Fig. 6.3.5.

We keep three lists for a time slot $i$: (1) *start* list in which bursts have their start time fall into this time slot and are sorted in ascending order of burst start time, (2) *finish* list in which bursts have their end time fall into this time slot and are also sorted in ascending order of burst end time, and (3) *full* list in which bursts use the full time slot span. Assume that the number of bursts in these three lists is $N_i^s$, $N_i^f$, and $N_i^f$, respectively. Given that $N_i$ is the total number of bursts placed in a time
slot $i$, we also maintain an attribute associated with this time slot, $N_i^* (\leq N_i)$, which is the maximum number of bursts out of these $N_i$ bursts that can be scheduled in this time slot. After admitting a burst with transmission time spanning from slot $i$ to $j$ ($i \leq j$), the bursts in these time slots should satisfy

$$N_k^* \leq W, \hspace{1em} \forall k \in \{i, \cdots, j\}.$$  \hspace{1em} (6.1)

If a burst can not be admitted, the burst is removed from the time slots in which it is placed and the burst is dropped.

We set the time slot span as the transmission time of a burst with minimum burst length $L_{\text{min}}$. We can easily see that a burst with length $L$ may occupy $\lceil \frac{L}{L_{\text{min}}} \rceil$ or $\lceil \frac{L}{L_{\text{min}}} \rceil + 1$ time slots. Admission control is divided into two steps: (1) pre-admission control and (2) time slot checking. In the first step, the incoming burst is attempted to schedule in the time slots it needs to use as implemented by Algorithm 3. If the first step fails, the node tries to rematch all the bursts in the time slots to check whether all the time slots have free resources to accommodate all the bursts after admitting a new burst $DB$ as shown in Algorithm 4.

A burst $DB$ placed in a time slot $i$ falls in one of the following three scenarios: (1) $DB$ uses the full time span of time slot $i$; (2) $DB$ has its start time fall into time slot $i$; and (3) $DB$ has its finish time fall into time slot $i$. Note that bursts in the start list conflict with all other bursts in the start list of this time slot, because transmission of a burst always takes no less than a time slot span. A similar observation is obvious for the bursts in the finish list. For all these three scenarios, Algorithm 3 checks whether Eq.(6.1) is satisfied.
In the first scenario, burst $DB$ conflicts with all other bursts in time slot $i$, and admitting burst $DB$ increases $N_i^*$. Therefore, Algorithm 3 only needs to check whether $N_i^* + 1 \leq W$ to verify whether $DB$ can be admitted in this time slot. The time complexity for this comparison is $O(1)$.

When a burst $DB$ is placed in the start (finish) list of time slot $i$, admission control tries to find free resources that can accommodate $DB$. Two cases needs to be considered: (1) $DB$ can be matched to a burst $B$ in finish (start) list, that is, $DB$ and $B$ are compatible; and (2) a wavelength is free in time slot $i$. If $DB$ can be accommodated with another burst $B$ on the same wavelength, we mark $DB$ and $B$ as matched. For an incoming burst $DB$ that falls into the start list of time slot $i$, admission control first searches for an unmatched burst $B$ in the finish list that is compatible with $DB$ and arrives the earliest. That is, we try to minimize the void between $DB$ and $B$. Similarly, when burst $DB$ is placed in the finish list, admission control searches for an unmatched burst $B$ in the start list to minimize the void. If such a burst $B$ can not be found, admission control checks whether $N_i^* + 1 \leq W$ to ensure that there is a free resource in time slot $i$ for burst $DB$. It takes $O(\log W)$ time to determine free resources for burst $DB$.

**Example 5** Assume that bursts $DB_0, \ldots, DB_3$ file into time slot 1 in Fig. 6.3.5. When $DB_1$ arrives, it is matched to burst $DB_0$. Burst $DB_2$ then finds a new wavelength for transmission. If $W = 2$, burst $DB_3$ can not be accommodated, because it conflicts with $DB_2$ and there is no free wavelength in time slot 1 according to Algorithm 3. However, if we rematch the bursts, i.e., $DB_0$ and $DB_1$ are matched with
When a burst $DB$ placed in time slot $i$ cannot find free resources in the pre-admission process stated above, Algorithm 4 re-matches the bursts in the start and finish lists to decide the number of wavelengths needed to schedule all the bursts in this time slot. Assume that the sorted start list $BS$ consists of bursts $BS_1, BS_2, \ldots, BS_{N_s}, \ldots, BS_{N_s}$, while the sorted finish list $BF$ maintains bursts $BF_1, BF_2, \ldots, BF_{n_e}, \ldots, BF_{N_s}$. For each burst $DB$ in start list, slot $i$ searches for the burst in finish list which finishes the latest and does not conflict with $DB$. If such a burst can be found, these two bursts can be scheduled using one wavelength. If the number of required wavelengths is no more than $W$, Algorithm 4 returns TRUE to indicate that slot $i$ can provide enough resources to accommodate all the bursts in slot $i$; otherwise, Algorithm 4 returns FALSE.

**Time and Space Complexity** In the admission control process, the most time-consuming part is Algorithm 4. In the worst case, there are $W$ bursts in the start list and finish list, respectively. Obviously, each burst in Algorithm 4 will be checked at most once. Therefore, the worst case time complexity for Algorithm 4 is $O(2W) = O(W)$ and Algorithm 3 has time complexity $O(W)$. Note that it takes $O(\log W)$ time to insert a burst into the start or finish list. Accordingly, the admission control process has time complexity of $O(W + \log W)$.

Assume $T$ is the time difference between the arrival time of the earliest arrival burst and the departure time of the latest departure burst among the bursts waiting for scheduling at a core node. The core node maintains at most $\left\lceil \frac{T}{L_{\min}} \right\rceil + 1$ time
slots. There are at most $2W$ bursts in each time slot. Given that $S$ is the space required to store the information of each burst, the space needed at the core node is $O(W \cdot S \cdot \lceil \frac{T}{L_{min}} \rceil)$.

**QoT Verification**

Burst transmission should also satisfy QoT requirement. After a burst is admitted in the previous free resource search step, QoT verification checks whether the PMD and OSNR constraints are satisfied if the burst is scheduled on the output link. The signal quality is unacceptable if at least one of these two constraints cannot be satisfied as shown in Fig. 6.3.6.

The PMD constraint is estimated via Eq. (5.1) using link state information, i.e., PMD parameters and link length information. If the PMD constraint cannot be satisfied, QoT verification fails; otherwise, the OSNR constraint is tested via Eq. (5.16) based on the gain and ASE factor parameters of amplifiers along the burst trans-
mission path. If the OSNR of the incoming burst is smaller than the pre-specified
\( OSNR_{\text{min}} \), the signal quality fails the QoT verification; otherwise, the resources de-
determined in the first step are verified to be acceptable in terms of signal quality.

**Deflection Routing**

If the primary route can not accommodate the incoming burst due to either resource
blocking or QoT blocking, the deflection route is retrieved and tried. In the case that
deflection routing also fails, the burst is dropped. On the other hand, if the available
resources satisfy the physical impairment constraints on the deflection route, the
burst is scheduled on the selected link, and the first control packet is updated and
forwarded.

\[ CP_1 \] needs be augmented to carry the cumulative PMD and OSNR along the
path traversed. Specifically, \( \sum_{k=1}^{H'} D_{PMD}^2(k) \times L(k) \) and \( NF_1 + \frac{NF_2-1}{G_1} + \frac{NF_3-1}{G_1G_2} + \cdots + \frac{NF_{M'-1}}{G_1G_2\cdots G_{M'-1}} \) are included in \( CP_1 \), where \( H' \) and \( M' \) are the number of links and the
number of amplifiers that the burst has traversed, respectively. When a burst can
be scheduled subject to the physical impairment constraints, these two cumulative
values are updated before \( CP_1 \) gets forwarded.

**Outgoing Wavelength Selection**

Each node stores a list of free channels (wavelengths) that are available for burst
scheduling in a \textit{free-channel queue} (FCQ). For a burst with service interval \([t_{sDB}, t_{eDB}]\),
the burst is scheduled onto the channel at the head of the FCQ at burst scheduling
time \( t_{RSV} \), and the channel is placed back into the FCQ at time \( t_{eDB} \). After outgoing
wavelength selection is completed, a second control packet containing the outgoing
wavelength information is sent to the downstream node.

6.4 Performance Evaluation

6.4.1 Simulation Setup

We evaluate the performance of our proposed algorithm by implementing the algorithms using ns-obs version 0.9a [122]. The network topologies used in the simulation are the NSF network topology (Fig. 5.4.5), 16-node Torus network topology (Fig. 5.4.6), and 8-node ring network topology (Fig. 5.4.7) with link length in kilometers. In these two networks, each link is bi-directional with a fiber in each direction. The number of data wavelengths and control wavelengths on each link are 8 and 2, respectively. The bandwidth of each wavelength is 10 Gb/s. An amplifier is applied every 100 km. The optical fibers can transmit light at about the speed of 200,000 km/s. The incoming self-similar traffic generated by OBS traffic generator is uniformly distributed between all pairs of edge nodes.

The parameters used in the simulations are as follows:

- Amplifier gain: 15 dB;
- ASE factor \( n_{sp} \): 1.5;
- \( D_{PMD}(k) \): 0.2 ps/(km)\(^{3/2} \);
- Fractional pulse broadening parameter \( \delta \): 0.1;
- \( OSNR_{min} \): 7.4 dB (BER = 10\(^{-9} \));
- Physical offset time: 20 ms;
• Functional offset time: 5 ms;

In Torus and ring network topologies, link PMD parameter $D_{\text{PMD}}(k)$ of the links with length 1000 km is $0.1 \, ps/(km)^{1/2}$.

The performance is measured by two metrics: burst blocking probability and average burst end-to-end delay. The simulation is run with different offered loads which are defined by Eq. (5.17).

In the simulation, we compare the performance of the impairment-aware algorithm in DOBS networks, the impairment-unaware algorithm in DOBS networks, and the impairment-aware algorithm in single header OBS networks. For the impairment-unaware algorithm in DOBS networks, the core nodes do not have knowledge about the QoT of bursts, and bursts are scheduled without deflection routing. That is, only one path is tried during the burst scheduling. At the destination, the signal quality is checked, and the burst is dropped if the signal quality is not satisfied. For the impairment-aware algorithm in single header OBS networks, LAUC-VF algorithm in the context of JET is used. At each node, the impairment-aware algorithms in both DOBS and single-header OBS networks schedule bursts for transmission by searching for available resources as well as verifying signal quality. If no QoT qualified free resource is available on the primary path, the deflection route will be checked.

6.4.2 Burst Blocking Probability

The burst blocking probability performance versus different offered loads is shown in Figs. 6.4.7 to 6.4.9. The simulation results show that the impairment-aware algorithm results in overall significantly reduced burst blocking probabilities than the
impairment-unaware algorithm in DOBS networks, since the impairment-aware algorithm takes into account physical impairment effects, and only bursts with quality transmission paths are scheduled. That is, unnecessary transmission of QoT-unqualified bursts can be avoided, which saves resources for burst transmission that satisfies the QoT constraints, and makes burst scheduling more effective. Furthermore, the impairment-aware scheduling algorithm deflects a burst onto an alternative path during scheduling to accommodate a burst if the burst can not be scheduled on the primary path either due to QoT blocking or resource blocking. Therefore, the blocking probabilities of the impairment-aware algorithm are reduced.

In all these three networks, the impairment-aware algorithm in DOBS networks achieves better blocking performance than its counterpart in single-header OBS networks. In single-header OBS networks, bursts are scheduled once they arrive at a node. In DOBS networks, control information is divided into two control packets.
to decouple resource request and resource allocation. The scheduling is delayed until functional offset time before bursts arrival. The delayed scheduling allows bursts to be scheduled in the order of the arrival time of bursts. This FCFS scheduling alleviates the voids problem on the transmission channels, which increases resource utilization and potentially improves blocking performance.

### 6.4.3 Average Burst End-to-End Delay

The simulation results for average burst end-to-end delay are depicted in Figs. 6.4.10 to 6.4.12. Generally the average burst end-to-end delay decreases as the network offered load increases. Bursts that need to traverse more hops in the network have a higher probability of being dropped. Consequently, bursts that traverse more hops constitute a smaller portion in the total number of bursts successfully received by the destinations at a higher offered load because more bursts compete for the limited
Figure 6.4.9: Burst blocking probability under different scheduling algorithms for ring network topology.

resources, and the bursts with more hops are more likely subject to burst drop.

In DOBS networks, the impairment-aware algorithm results in a larger average end-to-end delay than the impairment-unaware algorithm. When a burst can not be accommodated on the primary path due to either resource blocking or QoT blocking, it may still be scheduled onto the deflection route, which is longer than the primary path. Therefore, the average end-to-end delay for the impairment-aware algorithm is larger.

The end-to-end delay for the impairment-aware algorithm in DOBS networks is larger than its counterpart in single-header OBS networks. As shown before, more bursts can be scheduled with the impairment-aware algorithm in DOBS networks due to the better blocking performance of the impairment-aware algorithm in DOBS networks. End-to-end delay is potentially increased, because more bursts are accommodated using the impairment-aware algorithm in DOBS networks.
Figure 6.4.10: Average burst end-to-end delay under different scheduling algorithms for the NSF network topology.

6.5 Summary

In general, a physical impairment-aware scheduling algorithm is more complicated than a scheduling algorithm counterpart that does not consider physical impairments. In this chapter, we take advantage of a new signaling architecture DOBS to improve the time complexity of the impairment-aware OBS scheduling algorithm. Based on earlier work, we study the burst scheduling problem in OBS networks using two control packets for each data burst in order to decouple the resource request from resource reservation while taking into account physical impairment effects. We have proposed a burst scheduling algorithm which accommodates incoming bursts by primary path routing, deflection routing, and burst scheduling. We design an admission control mechanism to use network resources efficiently. At an OBS node, the proposed algorithm schedules bursts for transmission by searching for available resources as well
Figure 6.4.11: Average burst end-to-end delay under different scheduling algorithms for Torus network topology.

as verifying signal quality. Our simulation results demonstrate that the proposed algorithm is effective in terms of reducing the burst blocking probability.
Figure 6.4.12: Average burst end-to-end delay under different scheduling algorithms for ring network topology.
Algorithm 3 PreAdmissionControl

Require: The first control packet \( CP_i \) for a burst \( DB \)

Ensure: Return TRUE to admit \( DB \); otherwise, return FALSE

1: Decide the time slots \( i \) to \( j \) which \( DB \) needs to use;
2: Virtually add \( DB \) into these time slots;
3: admitted = TRUE;
4: for all time slot \( k = i, \cdots, j \) do
5: if \( DB \) uses the full span of time slot \( k \) then
6: if \( N_k^* + 1 > W \) then
7: admitted = FALSE;
8: else
9: incr \( N_k^* \);
10: end if
11: end if
12: if \( DB \) falls into the start list of time slot \( k \) then
13: if \( DB \) can be matched to an unmatched burst \( B \) in the finish list of time slot \( k \) then
14: Mark \( B \) and \( DB \) as matched;
15: else if a wavelength is free in time slot \( k \) then
16: incr \( N_k^* \);
17: else
18: admitted = FALSE;
19: end if
20: end if
21: if \( DB \) falls into the finish list of time slot \( k \) then
22: if \( DB \) can be matched to an unmatched burst \( B \) in the start list of time slot \( k \) then
23: Mark \( B \) and \( DB \) as matched;
24: else if a wavelength is free in time slot \( k \) then
25: incr \( N_k^* \);
26: else
27: admitted = FALSE;
28: end if
29: end if
30: if admitted = FALSE then
31: return FALSE;
32: end if
33: end for
34: return TRUE;
Algorithm 4 CheckSlot(i)

Require: Burst \( DB \) whose start or end time falls into time slot \( i \); \( N_i^* \leq W \)

Ensure: Return TRUE if slot \( i \) can provide enough resources to accommodate all
the bursts in slot \( i \); otherwise, return FALSE; \( N_i^* \leq W \)

1: \( newN_i^* = N_i^e + N_i^f + N_i^s \);
2: Mark all bursts in start and finish lists as unmatched;
3: \( n^e = 1 \);
4: \( n^s = 1 \);
5: while \( n^s \leq N_i^s \) and \( n^e \leq N_i^e \) do
6: \( \text{if end time of burst } BF_{n^e} < \text{ start time of burst } BS_{n^s} \) then
7: \( \text{decr } newN_i^* \);
8: \( \text{Mark } BF_{n^e} \) and \( BS_{n^s} \) as matched;
9: \( \text{incr } n^s \);
10: \( \text{end if} \)
11: \( \text{incr } n^e \);
12: \( \text{end while} \)
13: \( \text{if } newN_i^* > W \) then
14: Return FALSE;
15: else
16: \( N_i^* = newN_i^* \);
17: Return TRUE;
18: \( \text{end if} \)
Chapter 7

Physical Impairment Aware QoS Provisioning in Dual Control Packets Optical Burst Switched Networks

7.1 Problem Description

An important issue in OBS networks is to provide service differentiation in the optical layer because QoS support allows for the control of network traffic, which is the basis for sophisticated networking and also for charging, and thus sophisticated marketing strategies. Schemes developed for electronic networks can not be applied directly to OBS networks because of the following two reasons. The first reason is that the electronic buffer of bursts needs costly O/E/O conversion, resulting in the loss of data transparency. The second reason is that no random access memory in optical networks is available. Bursts are buffered via FDLs which can only delay the bursts for integer units of the FDL granularity. Hence, continuous delay can not be implemented.

Various schemes have been proposed to support QoS in OBS networks. An offset-time-based QoS scheme is proposed in [102,103] to assign extra offset time to the
higher priority classes without FDLs. The probability of successful resource reservation is increased with a longer offset time for higher priority bursts. The QoS scheme implemented in [104] isolates classes of traffic by assigning FDLs which function as optical buffer space, based on the Random Early Detection (RED) technique. An intentional dropping scheme is proposed in [105] in order to give a proportional burst loss probability for different service classes. A preemptive wavelength reservation mechanism is implemented in [106, 107] to provide different degrees of resource assurance to different classes of traffic in proportion to their service classes. QoS is provided in [80] by introducing prioritized contention resolution policies in the network core and a composite burst-assembly technique at the network edge, which resolves contention through prioritized burst segmentation and prioritized deflection, and the burst segmentation scheme allows high-priority bursts to preempt low-priority bursts. A generalized LAUC-VF algorithm is proposed in [108] to improve the QoS performance by prioritizing data bursts, maintaining multiple queues and utilizing limited optical buffer space.

All the QoS support schemes are integrated into burst scheduling algorithms. At the same time, physical impairment constraints have impacts on burst scheduling, e.g., path selection and routing, because paths may not satisfy the physical impairment constraints. In this chapter, we tackle the problem of service differentiation in OBS networks, taking physical impairment effects into consideration. We integrate a QoS provisioning scheme into burst scheduling in OBS networks that employ two control packets for each data burst. A high-priority burst requires a better quality of service in terms of blocking probability, and at the same time, the transmission of the burst
should satisfy its physical impairment constraints. To the best of our knowledge, currently there has been no previous work on QoS provisioning in OBS networks while taking physical impairment effects into consideration.

The rest of this chapter is organized as follows. Physical impairment-aware QoS supporting scheduling algorithm is described in Section 7.2. Section 7.3 reports the performance evaluation of the proposed scheme. Section 7.4 summarizes the related work. This chapter concludes in Section 7.5.

7.2 Impairment-Aware QoS Supporting Algorithm

Physical impairments considered are described in Section 5.2. In this chapter, we assume that OBS networks support two-class service differentiation. High priority bursts should experience a lower blocking probability than low priority bursts. We propose a QoS supporting algorithm based on the physical impairment-aware burst scheduling scheme that does not explicitly support QoS differentiation and is presented in Section 6.3.

We adopt a preemption approach to service differentiation problem in OBS networks. Same as in the previous chapter, we consider the problem in DOBS networks. The basic idea of this scheme is that a high priority burst can preempt a low priority burst upon contention. Burst preemption needs to satisfy two criteria: (1) no preemption occurs among bursts of the same priority, and (2) only one burst can be preempted at a time.

Upon the arrival of the first control packet of a burst, a network core node first
Figure 7.2.1: An example that shows resources are not available for high priority bursts in DOBS networks.

Attempts to perform admission control using the scheduling algorithm without QoS support which is proposed in Chapter 6. If the attempt succeeds, the burst is admitted to scheduling. If no QoT-qualified free resource is available for a high priority burst, the network core node tries to preempt a low priority burst so that the high priority burst can be accommodated.

**Example 6** In Fig. 7.2.1, before the arrival of high priority burst \(DB_6\), there are six low priority bursts \(DB_0\) to \(DB_5\). Assume that there are only three wavelengths for data burst transmission. \(DB_0\) and \(DB_3\) can be scheduled on the same wavelength. \(DB_2\) and \(DB_1\) can be matched, while \(DB_4\) and \(DB_5\) need to use a wavelength. These 6 bursts need 3 wavelengths for transmission. Therefore, there is no resource available to schedule burst \(DB_6\). In order to accommodate burst \(DB_6\), we can drop one of the six existing bursts so that the resource is available for burst \(DB_6\). In Fig. 7.2.1, three bursts \(DB_1\), \(DB_3\) and \(DB_5\) can be preempted.
Burst preemption has two steps. In the first step, a network core node searches for a low priority burst arriving later than the contending high priority burst. If more than one such burst can be found, the network core node preempts the burst with the earliest arrival time. If no low priority burst can be preempted to make resources available for the contending high priority burst in the first step, the network core node tries to preempt the low priority bursts arriving earlier than the contending high priority burst in the descending order of the burst finish time. If the previous two steps fail, the high priority burst is dropped due to resource blocking. The QoS supporting burst scheduling algorithm is summarized in Fig. 7.2.2.

![Flow diagram of the proposed QoS supporting burst scheduling algorithm](image)

Figure 7.2.2: Flow diagram of the proposed QoS supporting burst scheduling algorithm.

**Example 7** In Fig. 7.2.1, before the arrival of burst $DB_6$, bursts $DB_0$ to $DB_5$ need 3
data wavelengths. Assume that the number of data wavelengths is 3. If the incoming burst \(DB_6\) is of low priority, it is simply dropped or deflected because there is no resource available. Otherwise, burst \(DB_6\) needs to preempt one of the six bursts \(DB_0\) to \(DB_5\) to be scheduled.

First, the scheduling algorithm searches for a burst that arrives the earliest among the bursts which arrives later than the incoming burst \(DB_6\). In this case, there are two bursts, \(DB_1\) and \(DB_5\), whose arrival time is later than \(DB_6\). If \(DB_1\) and \(DB_5\) are of low priority, \(DB_1\) is selected to be preempted to make resources available for \(DB_6\) because burst \(DB_1\) arrives earlier than burst \(DB_5\). If \(DB_1\) is of high priority and \(DB_5\) is of low priority, \(DB_5\) is preempted. If both \(DB_1\) and \(DB_5\) are high priority bursts, the scheduling algorithm searches for a burst that finishes the latest among the bursts that arrive earlier than the incoming burst \(DB_6\). In Fig. 7.2.1, four bursts, \(DB_0\), and \(DB_2\) to \(DB_4\) arrive earlier than burst \(DB_6\). If \(DB_3\) is a low priority burst, it is preempted to schedule \(DB_6\). Otherwise, \(DB_6\) is dropped or deflected.

Note that QoT of resources determined in the scheduling process should be checked as described in Section 6.3. If a high priority burst can not be admitted on the primary route due to either resource blocking or QoT blocking, admission control with preemption and QoT verification are to be performed on the alternative route to check whether QoT qualified resource is available.

When a low priority burst is preempted, a second control packet is sent to the downstream node to release the pre-reserved resources for the burst. For each burst
that is admitted into the DOBS networks, the network core node selects an outgoing wavelength for the burst and then transmits the second control packet to the downstream node at functional offset time before the arrival of the data bursts.

7.3 Performance Evaluation

7.3.1 Simulation Setup

We evaluate the performance of our proposed algorithm by implementing the algorithms using ns-obs version 0.9a [122]. The network topologies used in the simulation are the NSF network topology (Fig. 5.4.5), 16-node Torus network topology (Fig. 5.4.6), and 8-node ring network topology (Fig. 5.4.7) with link length in kilometers. In these three networks, each link is bi-directional with a fiber in each direction. The number of data wavelengths and control wavelengths on each link are 8 and 2, respectively. The bandwidth of each wavelength is 10 Gb/s. An amplifier is applied every 100 km. The optical fibers can transmit light at about the speed of 200,000 km/s. The incoming self-similar traffic generated by OBS traffic generator is uniformly distributed between all pairs of edge nodes.

The parameters used in the simulations are as follows:

- Amplifier gain: 15 dB;
- ASE factor $n_{sp}$: 1.5;
- $D_{PMD}(k)$: $0.2 \text{ ps}/(\text{km})^{1/2}$;
- Fractional pulse broadening parameter $\delta$: 0.1;
- $OSNR_{min}$: 7.4 dB (BER = $10^{-9}$);
• Physical offset time: 20 ms;

• Functional offset time: 5 ms;

In Torus and ring network topologies, link PMD parameter $D_{PMD}(k)$ of the links with length 1000 km is $0.1 \text{ ps/(km)}^\frac{1}{2}$.

The performance is measured by two metrics: burst blocking probability and average burst end-to-end delay. The simulation is run with different offered loads which are defined by Eq. (5.17).

In the simulation, we compare the performance of our QoS provisioning algorithm in DOBS networks and the corresponding algorithm in single header OBS networks. For each algorithm, we compare the performance of high priority and low priority bursts. For the QoS provisioning algorithm in single header OBS networks, LAUC-VF algorithm with preemption in the context of JET is used. At each node, the algorithms in both DOBS and single-header OBS networks schedule bursts for transmission by searching for available resources as well as verifying signal quality. If no QoT qualified free resource is available on the primary path, the alternative route will be checked. A low priority burst may be preempted to make resource available for a high priority burst upon contention.

### 7.3.2 Burst Blocking Probability

The burst blocking probability performance versus different offered loads is shown in Figs. 7.3.3 to 7.3.5. The simulation results show that high priority bursts experience lower blocking probability than low priority bursts, because high priority bursts can preempt low priority bursts when no resource is available. The difference of the
Figure 7.3.3: Burst blocking probability under QoS provisioning scheduling algorithms for the NSF network topology in DOBS and single-header systems.

Blocking performance between high priority bursts and low priority bursts becomes more evident as the network load increases. This is because more bursts compete for network resource, and high priority bursts have better chances to reserve the required resources, because low priority bursts may be preempted by high priority bursts. At high offered loads, more low priority bursts are preempted to make resources available for high priority bursts.

In all these networks, the QoS provisioning algorithm in DOBS systems achieves similar blocking performance for high priority bursts to and better blocking performance than the QoS provisioning algorithm in single-header OBS networks. In single-header OBS networks, bursts are scheduled once they arrive at a node. In DOBS networks, control information is divided into two control packets to decouple resource request and resource allocation. The scheduling is delayed until functional offset time before bursts arrival. The delayed scheduling allows bursts to be sched-
Figure 7.3.4: Burst blocking probability under QoS provisioning scheduling algorithms for Torus network topology in DOBS and single-header systems.

uled in the order of the arrival time of bursts. This FCFS scheduling alleviates the voids problem on the transmission channels, which increases resource utilization and potentially improves blocking performance. In addition, this FCFS scheduling can better utilize the voids caused by preemption.

7.3.3 Average Burst End-to-End Delay

The simulation results for average burst end-to-end delay are depicted in Figs. 7.3.6 to 7.3.8. In general, the average burst end-to-end delay for high priority bursts is quite stable and the average burst end-to-end delay for low priority bursts decreases as the network offered load increases, although preemption may make the burst end-to-end delay performance curve fluctuate in the ring network. High priority bursts may preempt low priority bursts to find QoS qualified free resource. Low priority bursts which need to traverse more hops in the network have a higher probability of
being dropped. Consequently, bursts that traverse more hops constitute a smaller portion in the total number of bursts successfully received by the destinations at a higher offered load because more bursts compete for the limited resources, and the bursts with more hops are more likely subject to burst drop.

In all these networks, the QoS provisioning algorithm in DOBS systems results in similar average burst end-to-end delay performance for high priority bursts to and larger burst end-to-end delay for low priority bursts than the QoS provisioning algorithm in single-header OBS networks. As shown before, more low priority bursts can be scheduled with the algorithm in DOBS networks due to the better blocking performance of the algorithm in DOBS networks. End-to-end delay is potentially increased, because more bursts are accommodated using the QoS provisioning algorithm in DOBS networks.
Figure 7.3.6: Average burst end-to-end delay under QoS provisioning scheduling algorithms for the NSF network topology in DOBS and single-header systems.

7.4 Related Work

An important issue in OBS networks is to provide service differentiation in the optical layer because QoS support allows for the control of network traffic, which is the basis for sophisticated networking and also for charging, and thus sophisticated marketing strategies. We have briefly discussed QoS support in OBS networks in Section 2.5.

There are two major challenges faced by QoS enhancement in OBS networks [150]: (1) no RAM (beyond FDLs) in the core nodes to carry out scheduling, and (2) no feedback about network status to the edge nodes in case of one-way reservation.

In this chapter, we deal with the problem of relative QoS provisioning in terms of burst loss performance subject to the physical impairment constraints. The relative QoS support mechanisms in OBS networks can be classified into three categories: (1) offset-based, (2) segmentation-based, and (3) active-dropping-based.
Figure 7.3.7: Average burst end-to-end delay under QoS provisioning scheduling algorithms for Torus network topology in DOBS and single-header systems.

The QoS schemes for OBS networks proposed in [102–104] are offset-based. The basic rationale behind these schemes is to add an additional offset time between control headers and data bursts. The offset time assigned to a burst is varied based on the priority of the required service class of this burst. These offset-based schemes have the following characteristics:

- Bursts of higher priority classes have a longer waiting time for transmission;
- Higher priority bursts fragment wavelengths, and lower priority bursts try to fill the voids created by higher priority bursts;
- The schemes are non-preemptive.

Segmentation-based QoS support mechanisms are proposed in [80, 81] and evaluated in [151]. In these schemes, a burst can be divided into several independent
segments. Upon contention in the network, some segments in the contending burst are either discarded or deflected, while other segments are delivered to the destination. Consequently, the blocking performance is better than the non-segmentation-based schemes in terms of the lost bytes. Segmentation-based QoS provisioning has the following features:

- Segmentation is at the price of extra overhead introduced for every segment;
- The mean length of the bursts is smaller because the segments in the low priority burst may get lost;
- The scheduling complexity in the core nodes will increase because the core nodes need to decide what to do for every segment upon contention.

The Intentional Dropping scheme in [105] and Assured Horizon [150] are two examples of active-dropping-based QoS schemes. The active dropping policy acts as
a way for admission control, and it implements a burst dropper in front of each core node to drop lower priority bursts. In this way, the offered load at a core node can be controlled so that wavelengths will be available for higher priority bursts. The features of active-dropping-based QoS scheme are as follows:

- Lower priority bursts are intentionally dropped to ensure that wavelengths are available for higher priority bursts. However, higher priority class bursts may not arrive to use these resources, resulting in resource waste;

- Active dropping is non-preemptive;

- Burst loss probabilities do not depend on traffic characteristics, and realization can be simple;

- Isolation between classes can not be guaranteed. The overall burst loss probability will increase with the increase of lower priority traffic, which will result in increased blocking probability for all priority classes.

### 7.5 Summary

An important issue in OBS networks is to provide service differentiation in the optical layer. In this chapter, based on earlier work, we study the QoS provisioning problem in OBS networks that employ two control packets for each data burst, taking into account physical impairment effects. A high priority burst may preempt a low priority contending burst to guarantee service differentiation. We have proposed an physical impairment-aware QoS supporting burst scheduling algorithm which accommodates incoming bursts by admission control, preemption upon contention between high and
low priority bursts, outgoing channel selection, and signal quality verification. Our simulation results show that the proposed algorithm is effective in terms of providing service differentiation in OBS networks while considering physical impairment effects.
Chapter 8

Conclusions and Future Work

We conclude this dissertation by summarizing our contributions and identifying several new directions for future research on OBS networks.

8.1 Major Contributions

- Traffic Grooming in OBS Networks

  We have studied two traffic grooming problems in OBS networks: (1) per-hop traffic grooming, and (2) burst grooming by exploring node light-splitting capability.

  To reduce the switching overhead, small bursts may be groomed to reduce resource waste and switching penalty. We have studied the per-hop burst grooming problem where bursts with the same next hop may be groomed together, assuming all the network nodes have the grooming capability. Our objective is to minimize the number of formed larger bursts, and at the same time, to reduce the number of used wavelength converters (if wavelength conversion is available) and the number of used FDLs (if FDLs are available), that is, to strike a proper balance between burst grooming and grooming cost.
In order to reduce computation overhead and processing delay incurred at the core nodes, we assume that grooming can only be performed at edge nodes and the core node can send a burst to multiple downstream links, that is, the core node has light-splitting capability. We have attempted to groom small bursts into larger bursts, and select a proper route for each large burst, such that total network resources used and/or wasted for delivering the small bursts is minimized.

- Physical Impairment Aware Scheduling in Optical Burst Switched Networks

Optical signal transmission quality is subject to various types of physical impairment introduced by optical fibers, switching equipment, or other network components. The signal degradation due to physical impairment may be significant enough such that the bit-error rate of received signals is unacceptably high at the destination, rendering the signal not usable. Based on earlier work, we have studied the burst scheduling problem and proposed three effective burst scheduling algorithms in OBS networks, taking into account physical impairment effects. At an OBS node, the proposed algorithms schedule bursts for transmission by searching for available resources as well as verifying signal quality. Our simulation results show that the proposed algorithms are effective in terms of reducing the burst blocking probability.

- Physical Impairment Aware Ordered Scheduling in Dual Control Packets Optical Burst Switched Networks

Because the offset time of bursts varies in OBS networks, the voids or fragmen-
tation on the channels in the outgoing links can severely degrade the network throughput and blocking probability performance, if not dealt with carefully. A signalling architecture called Dual-header Optical Burst Switching (DOBS) was proposed by researchers to reduce the scheduling algorithm complexity. We have studied the burst scheduling problem and proposed an impairment-aware scheduling algorithm in DOBS networks. At an OBS node, the proposed algorithm schedules bursts for transmission by searching for available resources using admission control as well as verifying signal quality. Our simulation results demonstrate that the proposed algorithm is effective in terms of reducing the burst blocking probability.

- Physical Impairment Aware QoS Provisioning in Dual Control Packets Optical Burst Switched Networks

QoS provisioning is an important issue in OBS networks. However, schemes developed for electronic networks can not be applied directly to OBS networks because of two reasons: all optical data burst transmission and lack of optical RAM. We have dealt with relative QoS provisioning problem subject to the physical impairment constraints. A high-priority burst requires a better quality of service in terms of blocking probability, and at the same time, the transmission of the burst should satisfy the physical impairment constraints. A low priority burst may be preempted to make resource available for a high priority contending burst in our proposed algorithm. We have proposed a QoS-support burst scheduling algorithm which accommodates incoming bursts by
admission control, preemption upon contention between high and low priority bursts, outgoing channel selection, and signal quality verification. Our simulation results show that the proposed algorithm is effective in terms of providing service differentiation in OBS networks while considering physical impairment effects.

8.2 Future Work

We have addressed several important issues in OBS networks. Our work can be extended in a number of directions:

8.2.1 Physical Impairment-Aware Ordered Scheduling in DOBS Networks

We have proposed an efficient scheduling algorithm which takes physical impairments into consideration and makes use of two control packets to enable simple burst scheduling. Furthermore, we provide admission control during the burst scheduling process in order to improve network utilization. In the admission control process, we divide the time into time slots. A time slot transmits a burst of the minimum size.

Each time slot is associated with two sorted queues. One queue stores bursts which finish in the time slot, and the other maintains bursts which start in the time slot. We plan to improve the space complexity of the algorithm in order to reduce the space needed without increasing the time complexity of the algorithm.

A possible approach to decreasing the space complexity is to increase the length of the time slot. Each burst may fall into one or multiple time slots. A burst may start in a time slot, finishes in a time slot, or both. Each time slot maintains only
one queue of bursts. The bursts in a queue may be sorted according to the ascending order of burst arrival time or finish time.

Our objective is to perform admission control with the time complexity similar to or even less than the algorithm proposed in Chapter 6.

### 8.2.2 Impairment-Aware Proportional QoS Provisioning in Optical Burst Switched Networks

In Chapter 7, we have tackled the QoS provisioning problem in OBS networks, taking physical impairments into consideration. We assume that a high priority burst can preempt a low priority burst upon contention whenever a low priority burst is available to be preempted.

Another important service differentiation is to provide proportional differentiation, which can be described by

\[
\frac{q_i}{q_j} = \frac{s_i}{s_j} \quad (8.1)
\]

where \( q_i \) is the QoS metric and \( s_i \) is the differentiation factor for class \( i \). In this scenario, a low priority burst may not be preempted, as long as the proportional service differentiation can be provided based on Eq. (8.1).

We plan to design schemes that provide proportional QoS service differentiation in OBS networks, taking into consideration physical impairments. A challenging problem is that we need to provide proportional QoS service differentiation over both long and short time scales because it is important to provide proportional QoS service differentiation over a short time scale in networks with bursty traffic. At the same time, the transmission of the optical signal should satisfy physical impairment
8.2.3 TCP over OBS Networks

A key problem in OBS networks is contention resolution. Contention occurs when multiple burst competing for the same outgoing wavelength in the same link at the same time. Upon contention, only one burst can be accommodated and all the other bursts would be dropped, deflected, or delayed by FDLs. Bursts along a longer path are more likely to be dropped. This is unfair for bursts going through a longer path.

When TCP is implemented over OBS networks, the unfairness will result in lower throughput for TCP connections over longer path due to TCP’s congestion control mechanism. When a burst that contains TCP packets, such as ACK packets, is lost, the TCP congestion control mechanism will be triggered, which results in spurious retransmissions, reduces the TCP throughput and effective resource utilization.

A potential solution is to assign a high priority to bursts that carry TCP packets and at the same time, ensure in-order delivery of these bursts as much as possible by employing the DOBS signaling so that these bursts can be scheduled preferably at OBS network core nodes.
Bibliography


193


194


[121] R. E. Miller and editors J. W. Thatcher. *Complexity of Computer Computa-
tions, Reducibility among Combinatorial Problems*. Plenum Press, New York,
1972.


of Transmission Impairments on the Teletraffic Performance of Wavelength-
Routed Optical Networks. *Journal of Lightwave Technology*, 17(10):1713–1723,
October 1999.

Transmission Performance on Path Routing in All-Optical Transport Networks.


[127] I. Karamitsos. Analytical Model for Optical Burst Switching (OBS) Net-
works with Linear Transmission Impairments. *Proceedings of Second IFIP
International Conference on Wireless and Optical Communications Networks

[128] I. Baldine, P. Mehrotra, G. N. Rouskas, A. Bragg, and D. Stevenson. Intra-
and Inter-Domain Routing Architecture for Optical Burst Switched (OBS) Net-


European Conference on Networks and Optical Communications (NOC 2002),
Darmstadt, June 2002.

Optical Burst Switching. Proceedings of the SPIE Optical Networking and Com-
munications Conference (OptiComm 2002), Boston, MA, 4874:101–108, July
2002.
Publications

• Yuqi Fan and Bin Wang. Physical Impairment Aware Scheduling in Optical Burst Switched Networks. Accepted by Photonic Network Communications for publication.


• Yuqi Fan, Bin Wang, and Xubin Luo. Multicast Service Provisioning under a Scheduled Traffic Model in WDM Optical Networks, the third IASTED International conference on Communications and Computer Networks, Marina del Rey, CA, October 24-26, 2005.

• Bin Wang, Tianjian Li, Xubin Luo, Yuqi Fan and Chunsheng Xin. On Service Provisioning under a Scheduled Traffic Model in Reconfigurable WDM Optical