ABSTRACT

YANG, LI. Congestion Control and Quality-of-Service (QoS) on Jumpstart Optical Burst Switched Environment. (Under the direction of Professor George N. Rouskas.)

This thesis studies the congestion control and Quality-of-Service (QoS) problems in Optical Burst Switched (OBS) networks. It consists of three parts. In the first part, we consider path switching as a congestion control mechanism at the edge of the network. We study a suite of path-switching strategies, each of which gives a different method to estimate the path congestion online. We also develop a framework for combining several path switching strategies into hybrid strategies whose results are based on the decisions of multiple individual methods. We demonstrate the effectiveness and benefits of adaptive path selection via simulation.

In the second part of the thesis, we develop a general framework for absolute service guarantees for an OBS network in terms of the end-to-end burst loss. We first present a parameterized model for wavelength sharing. Then, we develop a heuristic for optimizing the policy parameters to support per-link absolute QoS guarantees. Finally, we present a methodology for acquiring the per-link parameters from the end-to-end QoS requirements so as to provide network-wide guarantees. We present numerical results to validate our approach.

In the third part, we present a per-link wavelength provisioning scheme based on Constrained Markov Decision Processes (CMDP) theory to provide service differentiation. Service differentiation is evaluated with two objectives on OBS networks: to maximize the constrained throughput; and to minimize the loss of the best effort traffic subject to the constraints on the priority traffics. The randomized threshold policies we obtain are simple to implement and operate, and make effective use of statistical multiplexing.

Congestion Control and Quality-of-Service (QoS) on Jumpstart Optical Burst Switched Environment

by

Li Yang

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

COMPUTER SCIENCE

Raleigh

2005

Approved By:

Dr. Rudra Dutta	Dr. Wenye Wang
Dr. George N Rouskas Chair of Advisory Committee	Dr. Khaled Harfoush

To my dear husband, **Wei**,
And my parents and sister,
Yanqing Yang, Yueli Pan, Xiaoke Yang,
for their endless love and support.

Biography

Li Yang, from Urumqi, China, earned a BS in Computer Engineering from Xi'an Institute of Posts and Telecoms in 1997 and an MS in Computer Science from Beijing University of Posts and Telecoms (BUPT) in 2000. She worked for one year as a software engineer in Alcatel Shanghai Bell-BUPT Research and Development Center in Beijing, China.

Li came to NCSU in 2001 as a PhD student in Computer Science program. Li has served as the Teaching Assistant for a series of Computer Science courses. Since 2002, Li worked with Professor Rouskas on congestion control and Quality-of-Service for Optical Burst Switched network. Li also worked as a software intern at Intel Raleigh Design Center.

After graduation, Li plans to pursue a career in research and development for networking.

Acknowledgements

This work is supported by the NSF under grant CNS-0322107 and by MCNC-RDI as part of the Jumpstart project. I am very grateful to members of my advisory committee, Dr. Rudra Dutta, Dr. Khaled Harfoush, Dr. George N. Rouskas, and Dr. Wenye Wang, especially to my advisor, Dr. George N. Rouskas for his support and invaluable guidance and suggestions.

Contents

List of Figures

Chapter 1

Introduction

With the huge amount of raw bandwidth available in fiber links brought by Wavelength Division Multiplexing (WDM) technology, and the rapid growth of transmission rates required by Internet traffic, harnessing the transmission capacity in optical fiber cost-effectively is essential for the development of the next generation optical Internet. There have been several different technologies developed for the transfer of data over WDM networks: broadcast-and-select networks [72], circuit switched wavelength routing [72], optical packet switching (OPS) [106], and optical burst switching (OBS) [71]. Broadcast-and-select networks, and circuit switched wavelength routing networks have been extensively studied, and deployed, while OPS is at the early stage of research and is not viable today. On the other hand, OBS networks have been deployed in testbeds [4] and are the subject of our work.

1.1 Optical Burst Switched Networks

OBS is a promising switching paradigm which aspires to provide a flexible infrastructure for carrying future Internet traffic in an effective yet practical manner. OBS separates the control (signaling) and data plane functions in the network in a way that exploits the distinct advantages of optical and electronic technologies. Signaling messages are processed electronically at every node in the network, while bursts are transmitted transparently end-to-end, without Optical-to-Electronic-to-Optical (OEO) conversion at intermediate nodes. Moreover, OBS transport is positioned between wavelength routing (i.e., circuit switching) and OPS. All-optical circuits tend to be inefficient for traffic that has not been groomed or statistically multiplexed, whereas OPS requires practical, cost-effective, and scalable implementations of optical buffering and optical header processing, which are several years away. OBS does not require buffering or packet-level parsing in the data path, and it is more efficient than circuit switching when the sustained traffic volume does not consume a full wavelength. The transmission of each burst is preceded by the transmission of a setup message (also referred to as burst-header control message), whose purpose is to reserve switching resources along the path for the upcoming data burst. An OBS source node does not wait for confirmation that an end-to-end connection has been set-up; instead it starts transmitting a data burst after a delay (referred to as "offset"), following the transmission of the setup message.

An OBS network is composed of users, optical switches (nodes) and fibers (shown in Figure 1.1). Users are devices, e.g., high-speed electronic routers or multiplexers, which generate optical bursts. An optical switch consists of two components: an optical cross-connect (OXC) (shown in Figure 1.2) which can optically forward a burst from an input to an output port without OEO conversion; and a signaling engine which processes signaling messages and controls the OXC switching fabric. Optical fiber links interconnect the network of switches, and also connect each user to one or more edge switches. A burst generated by a user travels past a series of fibers and switches in the OBS network, and terminates at another user.

The way in which resources are reserved has a significant effect on the performance of an OBS network, and is one of the most basic challenges in OBS. Several wavelength reservation schemes have been proposed for OBS, mainly including Just In Time (JIT) [95], Horizon [86] and Just Enough Time (JET) [71] [108] [81]. With these mechanisms, when the source node has a burst to transmit, it first sends a short request (SETUP) message to reserve the wavelength from the source to its destination. After reserving the wavelengths along the source-destination path, an acknowledgment message is sent back to the source, otherwise a failure message is returned to release the reserved wavelengths. It is noted that the source node need not wait for the acknowledgment and it will send out the burst right after an offset time, following the transmission of the SETUP message. The offset is selected such that each node on the path has enough time to process the request and

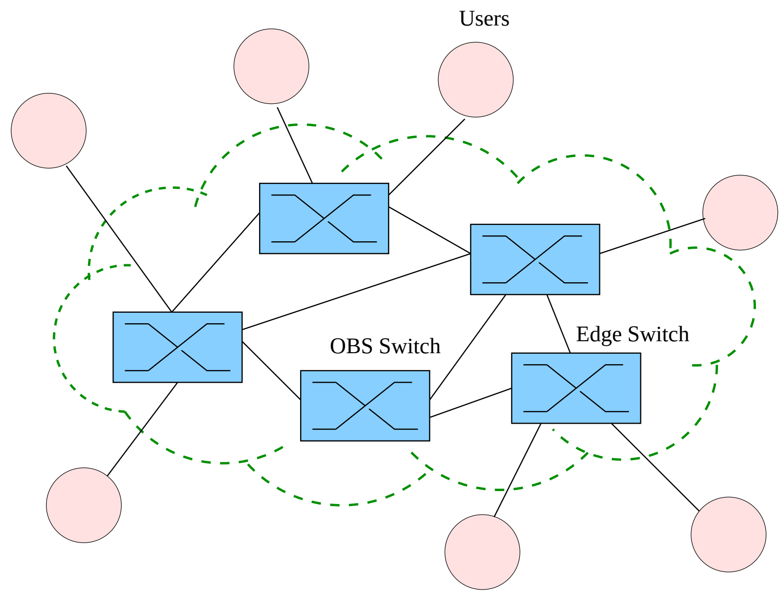

Figure 1.1: An OBS Network

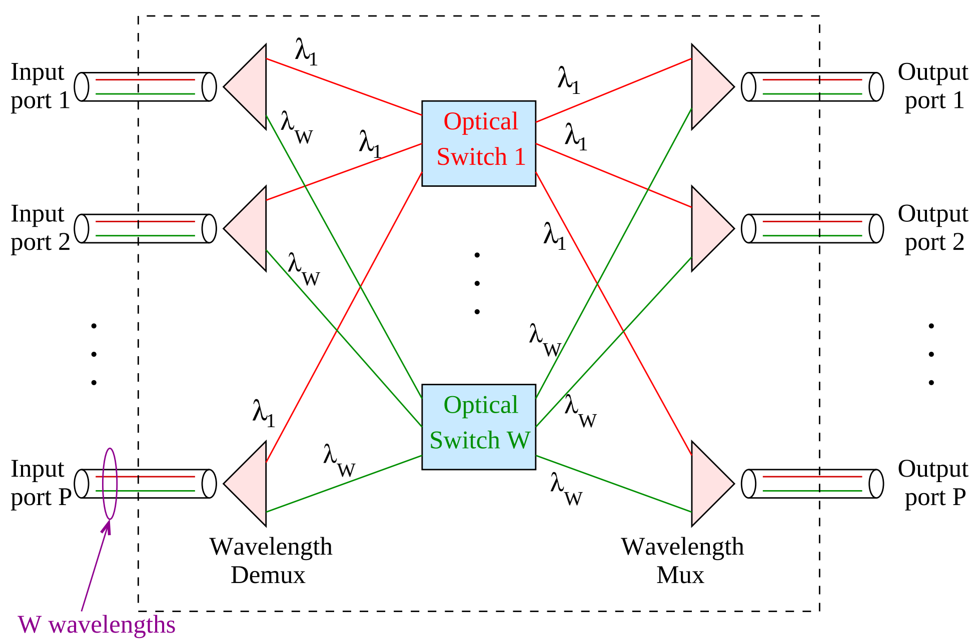

Figure 1.2: Optical Cross-Connect (OXC)

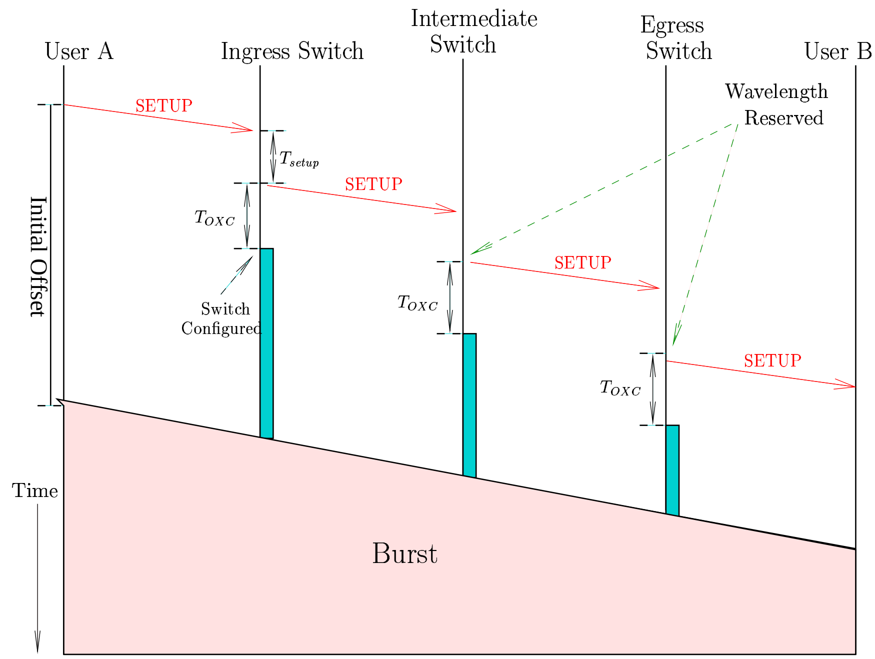

Figure 1.3: Operation of JIT (T_{setup}: processing time of SETUP message; T_{oxc}: OXC switching time

configure the switch. With JIT, the output channel is reserved for a burst immediately after the arrival of the corresponding SETUP message; the burst is rejected if no wavelength can be reserved for this burst at this time The immediate reservation mechanism of JIT is shown in Figure 1.3. On the other hand, the Horizon and JET are delayed reservation schemes, and are such that the reservation starts at the expected arrival time of the burst. Delayed reservation schemes can be further classified according to whether or not voids created by earlier SETUP messages are used for transmitting bursts whose SETUP messages arrive later. Void filling algorithms refer to the techniques that exploit the void intervals. Some delayed reservation schemes, such as Horizon, do not perform any void filling, and are very simple and fast. Horizon considers the horizon time for a given channel as the time after which no reservation is applied and so the next arriving burst can only reserve this channel after the horizon time. On the other hand, the JET protocol may lead to more efficient utilization of network resources by employing complex scheduling algorithms to serve bursts during unused time intervals. Although previous studies have shown that JET outperforms JIT in terms of burst loss probability, when the high processing overhead of JET is accounted and if the optical switch configuration time is much longer than the mean burst length, it is indicated in [84] that the simpler JIT reservation scheme appears to be a good choice for the foreseeable future.

In the next section, we explain in details the JIT reservation protocol we use in our work. It is worth mentioning that although we assume JIT in this thesis, the algorithms or policies we present and evaluate are independent of the specifics of the reservation protocol, and can be deployed alongside either the JET or the Horizon reservation schemes.

1.2 The `Jumpstart` JIT Reservation Protocol

In our work, we will assume that the OBS networks employs the JIT reservation protocol developed under the `Jumpstart` project [1]. The `Jumpstart` project was a collaborative effort of North Carolina State University (NCSU) and MCNC Research & Development Institute (MCNC-RDI). Supported by the Advanced Research and Development Agency (ARDA), the goal of the `Jumpstart` project was to create a signaling protocol and an associated architecture for a WDM burst-switching network. NCSU and MCNC-RDI have developed an open, published specification of the `Jumpstart` JIT signaling protocol [5, 6]. In November 2002, the JITPAC (Just-in-Time Protocol Acceleration Circuit) network controllers designed and built at MCNC-RDI were installed at three Advanced Technology Demonstration Network (ATD*net*) sites. The JITPAC network controllers implement the testbed's out-of-band signaling and control protocol [4]. We use the simulator that we developed as part of the `Jumpstart` project. The simulator accounts for all the details of the `Jumpstart` OBS signaling protocol, including all messages required for setting up the path of a burst and feedback messages from the network. Figure 1.4 illustrates the operation of the `Jumpstart` signaling protocol for a successful burst transmission. The connection is initiated with a SETUP message sent by the originator of the burst to its ingress switch. The SETUP ACK message by the ingress switch acknowledges the receipt of the SETUP message and informs the originating node which channel/wavelength to use when sending the data burst. After the offset time, the originator sends the burst on the indicated wavelength. If no blocking occurs on the path across the network, the SETUP message eventually reaches the destination node, which may choose to send a CONNECT message acknowledging the successful connection. Finally, the RELEASE message is sent by the originator to tear down the connection and release the reserved resources.

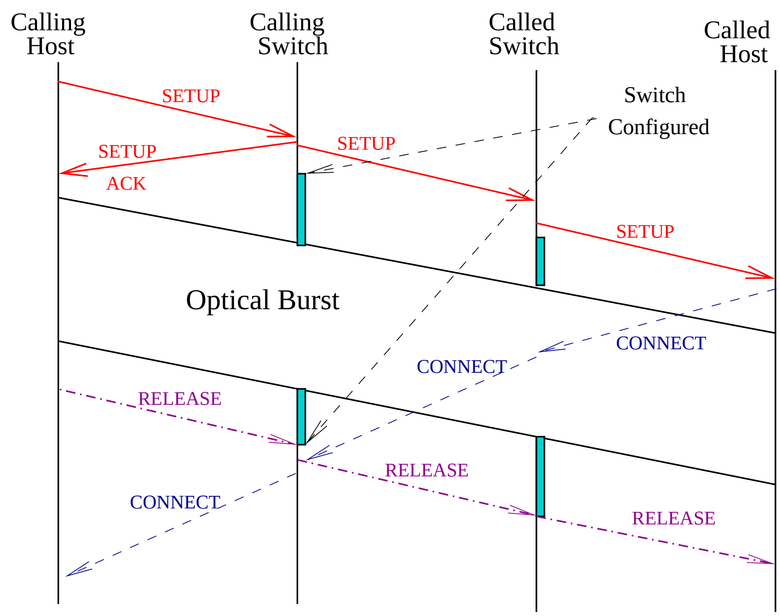

Figure 1.4: Jumpstart Signaling Protocol

1.3 Contributions

In this thesis, we study two important problems in OBS networks, namely, congestion control and the provision of absolute quality of service (QoS) guarantees. In the next two subsections we define these two problems and briefly discuss our approach to addressing them.

1.3.1 Congestion Control in OBS Networks

As defined in [51], congestion control is concerned with allocating the resources in a network such that the network can operate at an acceptable performance level when the demand exceeds or is near the capacity of the network resources. Due to increasing use of the networks, as well as due to increasing mismatch in link speeds caused by intermixing of old and new technology, congestion in computer networks is becoming a significant problem.

In OBS networks, because of the lack of optical buffers, a burst has to be dropped if the SETUP message cannot reserve an output port at any node along the source-destination path. Burst loss caused by transient or permanent overload is one of the most important issues in OBS networks. Furthermore, contention at the output channel is aggravated when the traffic becomes bursty or when the data burst duration varies and becomes longer. Appropriate contention-resolution mechanisms must be in place in an OBS network to manage the increased demand for resources during a period of congestion. Such mechanisms can be implemented either inside the network (i.e., at OBS switches) or at the source nodes where bursts originate. At the OBS switches, contention-resolution mechanisms can be employed to alleviate the effects of congestion. On the other hand, the congestion control techniques at the source avoid contention by buffering or dropping data at the source, in order to reduce the arrival rate into the network or by balancing the load in the network. In essence, the goal of all these congestion control mechanisms is to reduce the contention at the bottleneck links in the network [85].

In our work, we consider path switching as a congestion control mechanism at the edge of the network. Previous studies have shown in a different context that end-to-end path switching can bring performance benefits [80]. We investigate the concept of adaptive path selection in OBS networks and its potential to reduce the overall burst drop probability. We assume that each source maintains a (short) list of alternate paths to each destination, and uses information regarding the recent congestion status of the network links to rank the paths; it then transmits bursts along the least congested path. The continuous rankings of the paths are used to select the path that yields the best possible transmission conditions for the next to-be-transmitted burst. One main challenge of the appropriate path-switching mechanisms is to derive accurate estimates of congestion on each available path. We studied a suite of path-switching strategies, each of which gives a different method to estimate the path congestion in real-time. The set of path-switching strategies enable different levels of improvements in reducing burst drop probability. We demonstrate these improvements through a number of experiments under multiple network topologies and traffic patterns.

1.3.2 Absolute Quality-of-Service (QoS) Guarantees in OBS Networks

As OBS is becoming more widely accepted as a potential transport technology, supporting end-to-end Quality-of-Service (QoS) guarantees in OBS networks is arising as an important yet challenging issue. There is no standard definition of QoS. Here we adopt the notion from [58]: *QoS is network quality sufficient to satisfy user needs, however the needs may be expressed.* The actual QoS performance metrics of interest are projected to include such measures as loss, delay, and delay jitter guarantees. In general, there are two approaches to providing service differentiation and QoS [113]. In the *relative QoS model*, the service guarantees promised by the network provider to a given class of traffic are specified relative to the service guarantees of another class of traffic. Under the *absolute QoS model*, on the other hand, each priority class is guaranteed a worst-case service level (e.g., in terms of burst loss) that is independent of the service levels provided to other classes.

In this thesis, we study the problem of providing absolute QoS in OBS networks. Our objective is to maintain the burst drop rate for each guaranteed class below a given threshold. We put forward two mechanisms to satisfy the per-link blocking probability requirement of each priority traffic class. The first mechanism is inspired by earlier work on resource sharing [18, 54]. We first present a parameterized model for wavelength sharing among traffic classes that can provide a desired degree of isolation while taking advantage of statistical multiplexing gains. Then, considering a single OBS link, we developed a heuristic for optimizing the policy parameters to support per-link absolute QoS guarantees for a given set of heterogeneous traffic classes. The second mechanism is based on Constrained Markov Decision Processes (CMDP) for a single OBS link. The policy computed by CMDP can provide the optimal average class-based reward subject to the QoS requirements of the priority classes.

1.4 Structure of the Thesis

The thesis is organized as follows. In Chapter 2, we review research in OBS networks, with an emphasis on congestion control and QoS provisioning. Chapter 3 discusses the adaptive path switching problem. In Chapter 4 we present a framework for absolute QoS provisioning based on models for resource sharing. In Chapter 5, we develop a different mechanism for guaranteeing absolute QoS by combining ideas from admission control and Markov Decision Processes theory. We conclude our work and discuss future research directions in Chapter 6.

Chapter 2

Related Work

In this chapter, we first review several research topics in OBS networks; we then present a detailed survey of the research on congestion control and QoS problems in OBS.

2.1 Research Topics in OBS Networks

Since its introduction as a new switching paradigm for optical transport networks, optical burst switching has received significant attention from academic, vendor and even the operator point of view. With the continuous growth of research and development efforts in OBS networks over the last few years, a wide range of issues have been studied. This work can be broadly classified as follows.

1. *Network Architecure*
   Reference [86] proposed the idea of optical burst switching, and described a generic switch architecture together with its key performance and cost issues. References [10, 99] proposed an Internet architecture based on the optical burst switching mechanism and studied several important design issues of optical routers, such as issues in optical buffering and control. Reference [103] proposed and analyzed several access protocols for a WDM metro ring architecture with optical burst switching. Claiming that the transport layer architecture for OBS networks differs from transport architectures for other networks (e.g., IP), reference [9] studied the technical issues and

general requirements for a transport layer architecture (services and protocols) for OBS networks.

2. *Resource reservation schemes and protocols*

   As we discussed in Chapter 1.1, the resource reservation schemes have significant effect on the performance of OBS networks. As one of the most important challenges in OBS, resource reservation has received considerable attention and several reservation schemes have been put forward for OBS. Besides the JIT [95], Horizon [86], JET [71, 81, 108] schemes and the published specification of `Jumpstart` JIT signaling protocol introduced in Chapter 1.1, several others have been developed. A wavelength-routed OBS network was proposed in [30] which combines OBS with fast circuit switching and uses two-way reservations. Reference [64] proposed a "Forward Resources Reservation" method to reduce the burst delay at edge routers, which is also used to achieve burst delay differentiation. Finally, reference [63] analyzed a large set of online scheduling algorithms, indicating that the worst-case performance of an arbitrary best-effort online scheduling algorithm is closely related to a few factors, such as the range of offset time, the burst length ratio, scheduling strategy, and the number of data channels. Reference [63] also proposed a new reservation protocol, called Virtual Fixed Offset time (VFO), and it was claimed that VFO performs much better than JET.

3. *Contention resolution*

   In OBS networks, when one-way reservation scheme (e.g. JIT, JET, and Horizon) are adopted, it is possible that two bursts compete for the same output port, thus contention may arise. To resolve burst contention, many schemes have been introduced. For a discussion and comparison of various contention resolution mechanisms in OBS, please refer to Chapter 2.2.

4. *Burst assembly*

   In OBS networks, packets with the same destinations are usually assembled together using an assembly algorithm to reduce electronic processing load in the core nodes and increase switching efficiency [111]. Timer and burst length are two commonly used criteria in burst assembling. Reference [42] proposed a Fixed-Time-Min-Length burst assembly algorithm. The algorithm uses a fixed assembly time as the primary criteria and based on this primary criteria, it requires each burst size to be larger than a minimum length. On the other hand, the Max-time-Min-Max-Length burst assembly algorithm (proposed in [111]) uses the maximum assembly time as the primary criteria. In order to decrease the delay of an individual packet, it also allows a burst to be sent out as soon as the burst length reaches or exceeds a given maximum burst length. Reference [111] also demonstrated that after the assembly, the traffic will in general approach the Gaussian distribution. References [89] and [91] addressed the issue of providing differentiated services to IP packets over OBS networks via collecting packets into bursts that have the same QoS requirements. Having studied the performance of TCP traffic in OBS networks, reference [11] proposed the Adaptive-Assembly-Period (AAP) algorithm for assembling TCP traffic and indicated that burst assembly mechanisms affect the behavior of TCP in that the assembled TCP traffic becomes smoother in the short term, and more suitable for transmission in optical networks.

5. *Network model and performance evaluation*

   There are a lot of publications on modeling and evaluating the performance of OBS networks. References [12, 24, 25, 86, 87, 96, 109] studied the performance of OBS networks in terms of burst blocking probability. Under the assumption of no buffering and Poisson arrivals, references [12,24,25,87] analyzed an output port of an OBS node; while reference [74] evaluated the performance of the whole network. Following up the study in reference [74], [112] analyzed the burst blocking probability of the OBS networks when deflection routing is available. References [86, 109] studied the system when buffering is provided in OBS. In contrast to previous work, reference [96] modeled a system having multiple classes of bursts. In addition, references [62, 101, 102] studied the system with burst arrival distributions other than Poisson process. Finally, references [2, 73] studied the case of no and limited wavelength conversion, respectively.

Several other interesting research fields in OBS networks, include protection and restoration [43,59,60], multicasting in OBS [52,53], TCP over OBS networks [11,22,110], and scheduling algorithms over OBS networks [61, 67, 86, 99, 100, 105]. For a more complete description of the recent works on OBS networks please refer to [15].

2.2 Congestion Control in OBS Networks

As discussed in [51], the congestion problem can not be solved by simply adding buffer space, high-speed links, or high-speed processors. In essence, congestion control is not a trivial problem because of the number of requirements, such as low overhead, fairness, responsiveness, and so on. Despite these difficulties, congestion control is an important problem [41, 51, 55, 66], that must be addressed to ensure that networks do not collapse under overload conditions.

Because of its importance, the congestion control in OBS networks has drawn a lot of attention in recent years and several mechanisms have been put forward. Some of them can be implemented at the edge nodes, and some can be deployed inside the network. In this section, we present a survey of related work on congestion control in OBS networks. For a general survey of congestion control schemes on other network types, the reader is referred to [46, 98].

2.2.1 Congestion Control Mechanisms Within the OBS Networks

Contention-resolution schemes in OBS networks can be based on one of four orthogonal approaches, or a combination thereof: buffering, wavelength conversion, burst segmentation, or deflection [88, 107]. All these approaches require additional hardware or software components at each OBS switch, increasing their cost significantly; furthermore, practical implementations of some of these components require technology which may be several years from maturity. For instance, optical buffering solutions based on fiber delay lines are not currently cost-effective or scalable. Similarly, although wavelength conversion has been shown to be quite effective in alleviating output port contention [107], wavelength converters are expensive and complex devices, and this state of affairs is expected to continue in the foreseeable future.

Burst segmentation [88] refers to the process of discarding parts of a burst which overlap with another burst at the output port of an OBS switch. Burst segmentation has been proposed to improve the data loss due to contention [88], as well as a mechanism for providing differentiated service in an OBS network [90]. But segmentation also faces technological challenges, such as the ability to optically detect segments in a burst, to accurately truncate a burst in a way that the remaining data can be recovered at the receiver, and to signal downstream nodes of the reduced burst length. As a further step from segmentation, there are several policies which select the whole burst(s) to drop to resolve contention. In [35], Farahmand described the hardware implementation of the shortest-drop policy (SDP). When contention is detected, the SDP drops the lower priority burst with the shortest duration and latest arrival time. The look-ahead Window Contention Resolution (LCR) in [34] constructs a look-ahead window (LaW) with a size of W time units and decides which incoming bursts should be dropped based on a collective view of multiple stored SETUP messages. It is noted that LCR needs the support of FDLs at each hop to delay data bursts by W time units and it may increase the total end-to-end burst delay.

Deflection routing [7] is another mechanism that can be used to reduce the burst loss due to output port contention. In this approach, each switch maintains several paths to a destination, with one path designated as primary (default). When the primary path of an incoming burst is not available, the switch deflects the burst to one of the secondary paths. A deflection routing protocol for OBS networks was proposed in [93], while [48, 112] analyzed the performance of deflection routing. However, deflection routing in OBS networks has several disadvantages. A practical implementation would require intermediate switches which deflect a burst to somehow increase its offset, an operation that is impossible without the use of buffers (alternatively, each burst must have an offset large enough to account for all possible deflections in its path, severely degrading the performance of the network). When deflection decisions are made at each switch without coordination with the rest of the network (a typical approach given the limited amount of time between the SETUP message and the data burst), there is great potential for routing loops which can have disastrous effects in an optical network [49]. Finally, deflection routing is by nature suboptimal since it only considers the congestion of the current switch, not the state of the links further along the path; and it may cause undesirable vibration effects, as explained in [112].

2.2.2 Congestion Control Mechanisms at the Edge Nodes

It is also possible to employ a different set of mechanisms at the edge of the network (i.e., at ingress nodes or the burst sources) to reduce the level of burst contention inside the network. These approaches typically do not require or require only minimal support from the signaling protocol (e.g., feedback regarding the status of burst transmissions), and can be used when it is desirable to simplify the architecture of core OBS switches to improve their scalability and contain the overall network cost. Alternatively, edge and core nodes may each implement their own mechanisms and coordinate to further reduce burst loss. In general, the edge node strategies can be implemented in one of two ways: shaping or regulating source traffic, or balancing the load via appropriate routing algorithms. The two methods may also be combined for a more effective solution.

One way to perform the traffic shaping is through a burst assembly mechanism such as those proposed in [89], [111], [11], [69]. A threshold-based burst assembly scheme was proposed in [89], and in conjunction with a burst segmentation policy it can provide QoS in OBS networks. In [111], the statistical properties such as the burst length distribution, inter-arrival time distribution, as well as correlation structure of assembled burst traffic from several burst assembly algorithms were studied. A study of the effect of assembly algorithms on TCP traffic was carried out in [11]. Finally, the authors in [69] introduced a new burst assembly algorithm which can change the data burst sizes adaptively in accordance with the offered load and thus offers high average data burst utilization. A new switching architecture for IP/WDM was put forward in [31], which combines proactive channel reservation with periodic shaping of ingress-egress traffic aggregates. The common point of the above mechanisms is that they do not need feedback from the OBS networks. As proposed in [92], if feedback about the network condition is provided, a modified TCP decoupling approach can be used in OBS networks to congestion-control the traffic load offered to an OBS switch and regulate the timing of sending bursts. A source flow-rate control (SFC) technology is presented in [36]. SFC dynamically adjusts the data burst transmission rate at edge nodes in response to core nodes' feedback signals. In addition, Detti in [22] studied the relationship between the burst aggregation time and the TCP Reno send rates in an OBS IP optical network.

There are also several schemes to achieve congestion control by balancing the load in the network, and they mainly make use of one of these techniques: wavelength assignment, route calculation, route selection or a combination of two or more of these techniques. One edge node strategy that has the potential to improve burst contention significantly, especially when wavelength conversion is unavailable or sparse, is wavelength assignment. A priority wavelength selection algorithm was presented in [94], and a comprehensive study of wavelength selection strategies for OBS networks can be found in [82]. An algorithm for buffering bursts electronically at edge nodes and scheduling them to prevent burst overlap at links inside the network was proposed in [62]. A traffic engineering approach to select paths for source routing so as to balance the traffic load across the network links was investigated in [83]. Finally, a dynamic scheme for selecting routes at the burst sources was proposed in [85]. Each source maintains a (short) list of alternate paths to each destination, and uses link congestion information to rank each path. The source uses the least congested path to transmit its bursts. We note that a similar technique, referred to as "end-to-end path switching" was proposed and evaluated recently for selecting one among a set of Internet paths [80]; the main finding of the study was that path switching can result in substantial improvement in packet loss.

In our study, we undertake a comprehensive study of adaptive (end-to-end) path selection in OBS networks. Our objective is to develop a methodology for sources to dynamically switch traffic between a predetermined set of paths in a way that minimizes the overall burst drop probability in the OBS network. This work is presented in Chapter 3.

2.3 Guaranteeing Quality of Service (QoS)

As the future high speed networks are expected to carry a wide range of traffic types, providing a good quality of service to the large variety of users is an important yet challenging task [58]. Due to its importance and difficulties, there have been extensive research on the architectures for providing IP service differentiation in the last few years. Generally there are two different approaches for service differentiation: integrated services and differentiated services. The integrated services (IntServ) approach [97] focuses on individual packet flows, and *admission control, packet forwarding mechanisms,* and *Resource Reservation Protocol* (RSVP) are the three major components of IntServ. However, as discussed in [79], the IntServ architecture makes the management and accounting of IP networks significantly more complicated. The DiffServ approach was proposed as an architecture with simple mechanisms inside the network [32, 33]. With the DiffServ model, packets are classified prior to entering the network via a *packet marking mechanism*, and the services that a router insider the network provides to a packet are solely dependent on the packet's class. The internal router can treat packets according to *threshold dropping* or *priority scheduling* policy; while packet marking mechanisms can be one of the two forms, *edge marking* and *edge dropping.* Several mechanisms have been proposed for the packet marking and router mechanisms [20, 21, 40, 44, 47, 50, 77, 78]. For a comparison of the performance from different combinations of these various mechanisms, please refer to [76]. As we explained earlier in Chapter 1.3.1, there are generally two approaches, the relative QoS and the absolute QoS, to provide DiffServ. (A detailed comparison of these two approaches can be found in [26].) The proportional service differentiation model was introduced in [27] and [29] and it can provide quantitative service difference between classes as an effort to strengthen the old relative QoS model. The proportional differentiation model states that certain class performance metrics, such as the per-hop queueing delays and packet drops, should be proportional to the corresponding differentiation parameters that are chosen according to the pricing or policy requirements. In [28], a dynamic class selection framework was proposed and it can be used with the proportional QoS differentiation approach to provide absolute QoS. Recently Chen reviewed recent research efforts on the proportional service model and gave implementation strategies for providing various QoS metrics in [14]. The authors also discussed the end-to-end and per-link approaches to achieve the absolute service bounds within the proportional service model.

Most of the recent research in QoS over OBS networks has been focused on providing relative service differentiation, and a variety of schemes have been proposed, such as assigning an additional offset to higher priority bursts [109], intentionally dropping non-compliant bursts [13], mapping different classes of packets into bursts of different priorities [89, 91], selectively choosing a burst to segment or deflect during contention [90], and allowing in-profile bursts to preempt out-of-profile ones [65]. The OBS core in [89–91] needs to support burst segmentation to resolve contention. In the extra offset scheme [109], additional offset time is given to high priority bursts, thus the probability of successful resource reservation for high priority traffic is increased. However, as mentioned in [13, 70] this method suffers from unfavorable end-to-end delay and unfairness (increased blocking probability) for long bursts of low priority traffic. As an effort to acheiving proportional service differentiation in OBS networks, the scheme proposed in [13] quantitatively adjusts the blocking probability p_i of class i so that it can be proportional to its differentiation factor τ_i. Basically the following relationship is maintained for every traffic class,

$$\frac{p_i}{p_j} = \frac{\tau_i}{\tau_j}. \tag{2.1}$$

Each switch node tracks the numbers of burst arrivals and dropped bursts for all traffic classes, and employs an intentional burst dropping scheme to maintain the proportional differentiation equation (2.1). A study of absolute QoS guarantees in OBS networks can be found in [113], where two mechanisms were proposed to enforce a loss probability threshold for each guaranteed traffic while reducing the loss rate of the non-guaranteed traffic: an early dropping mechanism to selectively drop the non-guaranteed traffic, and a wavelength grouping strategy to allocate the necessary wavelengths to each priority traffic. Finally, the study in [64] differs from the above in that it considers delay, rather than burst drop probability, as the QoS parameter to be guaranteed.

We will study the problem of providing absolute QoS guarantees in OBS networks and present a general framework for absolute service guarantees to users of an OBS network in terms of the end-to-end burst loss in Chapter 4 and Chapter 5. In the next section, we will discuss our proposal to the contention resolution problem in OBS networks.

Chapter 3

Adaptive Path Selection in OBS Networks

In this chapter, we will undertake a comprehensive study of adaptive (end-to-end) path selection in OBS networks. Our objective is to develop a methodology for sources to dynamically switch traffic between a predetermined set of paths in a way that minimizes the overall burst drop probability in the OBS network.

The chapter is organized as follows. In chapter 3.1 we discuss our assumptions regarding the OBS network we consider in our study. In Chapter 3.2 we describe pure path switching strategies each utilizing partial information about the network state to select one of a set of available paths to route bursts. In Chapter 3.3 we develop a framework for combining several path switching strategies into hybrid (or meta-) strategies which base their routing decisions on the decisions of multiple individual methods. In Chapter 3.4, we present simulation results to demonstrate the effectiveness and benefits of adaptive path selection.

3.1 The OBS Network Under Study

We will use $G = (V, E)$ to denote an OBS network. V is the set of switches, $N = |V|$, and $E = \{\ell_1, \ell_2, \ldots, \ell_M\}$ is the set of unidirectional fiber links, $M = |E|$. Each link in the network can carry burst traffic on any wavelength from a fixed set of W wavelengths, $\{\lambda_1, \lambda_2, \cdots, \lambda_W\}$. We assume that each OBS switch in the network has full wavelength conversion capabilities which are used in the case of wavelength contention. The network does not use any other contention resolution mechanism. Specifically, OBS switches do not employ any buffering, either electronic or optical, in the data path, and they do not utilize deflection routing or burst segmentation. Therefore, if a burst requires an output port at a time when all wavelengths of that port are busy transmitting other bursts, then the burst is dropped.

The OBS network employs source routing, in that the ingress switch (source) determines the path of a burst entering the network. The path over which the burst must travel is carried by the SETUP message that precedes the transmission of the data burst. We assume the existence of a routing algorithm that is capable of computing a set of k alternate paths for each source-destination pair; the number k of such alternate paths is relatively small, i.e., $k = 2 - 4$. Each source node maintains the list of paths for each possible destination, and is responsible for selecting the path over which a given burst will travel. Once the source has made a routing decision for a burst, the path is recorded in the SETUP message and it cannot be modified by downstream nodes (i.e., no deflection is allowed).

All source nodes use the same *path switching strategy* to make routing decisions on a per-burst basis. A path switching strategy is characterized by the metric used to rank the paths to a certain destination node. In general, the metric is designed to reflect the likelihood that a burst transmitted on a particular path will experience resource contention and be dropped before it reaches its destination. Whenever a new burst is ready for transmission, the source node selects the "best" path according to the metric used (with ties broken arbitrarily) and injects the burst into the network. We present a number of path switching strategies, and their associated metrics, in the next section.

The rank of each path maintained at a source node is updated dynamically based on information regarding the state of the network collected by the node. We assume that the control plane of the OBS network provides support for the collection and dissemination of information required by the path switching strategies. For instance, this information may be part of the feedback the source receives from the signaling protocol regarding the success or failure of each burst transmission; the `Jumpstart` just-in-time (JIT) signaling protocol was designed to provide such feedback [5]. Alternatively, the OBS switches may collect information and statistics regarding the (long-term) congestion status of their links, and use a link-state protocol to disseminate this information to the rest of the network. Since signaling and state dissemination protocols are required for a variety of network functions, the additional overhead due to the path switching strategies we propose in this paper is expected to be only moderate.

As network dynamics change due to shifts in traffic conditions, congestion levels, and the actions of burst sources, information about these changes will be propagated to the edge of the network using the mechanisms discussed above. In turn, the path rankings at the source nodes may be updated to reflect the new state of the network. As a result of adaptive path selection, different bursts between a certain source-destination pair may take different paths through the network. However, we assume that path switching takes place at large time scales relative to the burst transmission (and generation) times. In particular, the rate of path switching depends on network constants, such as the network diameter and the parameters of the state dissemination protocol (e.g., update interval), whose values are in the hundreds of milliseconds (or even seconds). Consequently, at steady state, a (relatively large) number of successive bursts will use the same path before path switching takes place.

In the next section, we introduce the concept of pure path switching strategies, and present a set of such strategies in detail.

3.2 Pure Path Switching Strategies

A path switching strategy uses information about the current state of the OBS network to select one of a small number of routing paths for transmitting burst traffic between a source-destination pair. There are several different pieces of information that could be used to describe the congestion level in the network (for instance, link utilization, end-to-end path burst drop rate, etc.); and there are several ways in which this information can be combined into a metric to rank paths. It is unknown which types of information or what metrics perform best for path switching in terms of burst drop probability. In this section we present a suite of *pure* path switching strategies, i.e., strategies which use a single path selection method. In the next section, we introduce a framework for combining several path selection methods into hybrid strategies which provide for further improvement in performance.

3.2.1 Weighted Bottleneck Link Utilization (WBLU) Strategy

The weighted bottleneck link utilization (WBLU) strategy ranks the candidate paths using information on link utilization. The motivation behind this strategy is to reduce or prevent contention by using paths with less utilized links.

Consider a (directional) link ℓ of the OBS network, let $Succ(\ell, t)$ denote the set of bursts that have successfully traversed link ℓ until time t, and let T_i denote the length of burst i. The utilization $U(\ell, t)$ of link ℓ at time t is defined as:

$$U(\ell, t) = \frac{\sum_{i \in Succ(\ell,t)} T_i}{Wt} \tag{3.1}$$

where W is the number of available wavelengths; at time $t = 0$, we assume that the utilization $U(\ell, 0) = 0$ for all links ℓ.

Consider now a source-destination pair (s, d), and let $\{\pi_z, z = 1, \cdots, m\}$ be the set of m candidate paths for transmitting bursts from node s to node d. Let $\{\ell_k, k = 1, \cdots, |\pi_z|\}$ be the set of links composing path π_z which has length (in number of hops) $|\pi_z|$. At time t, the WBLU strategy routes bursts from s to d along the path $\pi_{z^\star(t)}$ whose index $z^\star(t)$ is obtained using the following metric:

$$z^\star(t) = \arg \max_{1 \le z \le m} \frac{1 - \max_{1 \le k \le |\pi_z|} U(\ell_k, t)}{|\pi_z|} \tag{3.2}$$

The numerator in the above expression is the available capacity of the bottleneck link in a given path π_z. Therefore, the WBLU strategy routes bursts along the path with the highest ratio of available bottleneck link capacity to path length. By taking the number of hops into account as in expression (3.2), we ensure that if the bottleneck link utilization is similar for two paths, then the shortest path is selected for routing; the longer path is preferred only if the utilization of its bottleneck link is significantly lower than that of the shorter one. We note that a similar metric for ranking paths was used in [19] as part of a routing and wavelength assignment algorithm for wavelength routed networks.

We note that it is possible to either reset the utilization values periodically or not. The latter approach is easier to implement, but it is not adaptive to load fluctuation. Resetting the values periodically provides better performance under dynamically changing traffic since they more accurately reflect the recent state of the network. In this case, the length τ of the update period is an important parameter that must be carefully selected to balance a set of conflicting requirements. A small update interval period may cause path oscillations and lead to unstable network behavior, while with a very long interval this strategy may fail to react to changing demands in a timely manner. In general, the length of the update interval will be a function of the diameter of the network and the specific algorithm used to disseminate the link congestion information. For a discussion of the factors that need to be considered in selecting the update interval, the reader is referred to [85].

Recall that we have made the assumption that shifts in traffic demands take place at longer time scales than the operation of the path switching strategy. Therefore, the update interval is taken to be longer than the time scales we consider in this work, during which no changes in traffic patterns are considered. Selecting an appropriate value for the update interval under changing demands is outside the scope of our work.

3.2.2 Weighted Link Congestion (WLC) Strategy

The objective of the weighted link congestion (WLC) strategy is to route bursts along the path that is most likely to lead to a successful transmission. To this end, the source uses information on link congestion along each path to infer the burst drop rate of the path. This strategy assumes the existence of a link-state protocol that disseminates information on link congestion.

Let $N_{succ}(\ell, t)$ (respectively, $N_{drop}(\ell, t)$) denote the number of bursts that have been successfully transmitted along (respectively, dropped at) link ℓ up to time t. We define the congestion level $c(\ell)$ of link ℓ at time t as the fraction of bursts that have been dropped at the link:

$$c(\ell, t) = \frac{N_{drop}(\ell, t)}{N_{drop}(\ell, t) + N_{succ}(\ell, t)} \tag{3.3}$$

We assume that at time $t = 0$, the congestion $c(\ell, 0) = 0$ for all links ℓ.

Let π_z be a candidate path for routing bursts between a source-destination pair (s, d), consisting of links $\ell_1, \cdots, \ell_{|\pi_z|}$. Assuming that link drop probabilities are independent, at time t the probability that a burst will be dropped along this path can be calculated as:

$$b(\pi_z, t) = 1 - \prod_{1 \le i \le |\pi_z|} (1 - c(\ell_i, t)) \tag{3.4}$$

The weighted link congestion (WLC) strategy routes bursts from s to d along the path $\pi_{z^\star(t)}$ whose index $z^\star(t)$ is obtained using the following metric:

$$z^\star(t) = \arg \max_{1 \le z \le m} \frac{1 - b(\pi_z, t)}{|\pi_z|} \tag{3.5}$$

As in expression (3.2), this metric takes the number of hops of each path into account, in order to ensure that longer paths are preferred over shorter ones only when they offer a substantial improvement in drop probability.

3.2.3 End-to-End Path Priority-based (EPP) Strategy

The end-to-end path priority-based (EPP) strategy is similar in spirit to the WLC strategy in that it also attempts to route bursts along paths with low drop probability. However, rather than relying on information on individual link congestion levels to infer the burst drop probability, this strategy requires the source to directly measure this probability from feedback messages it receives from the network regarding the status of each burst transmission.

Consider the source-destination pair (s, d), and let π_z be one of the m candidate paths for this pair as before. Let $N_z(t)$ denote the total number of bursts that have been transmitted (successfully or unsuccessfully) from s to d on path π_z up to time t. The EPP strategy assigns a priority $prio(\pi_z, t)$ to path π_z at time t which is updated each time a new burst is transmitted on this path, and is recursively defined as:

$$prio(\pi_z, t) = \begin{cases} 1.0, & t = 0 \\ \frac{prio(\pi_z, t-1) \times N_z(t-1) + 1}{N_z(t-1) + 1}, & \text{burst transmitted successfully at time } t \\ \frac{prio(\pi_z, t-1) \times N_z(t-1)}{N_z(t-1) + 1}, & \text{burst transmitted unsuccessfully at time } t \end{cases} \tag{3.6}$$

$N_z(t)$ is also updated as: $N_z(t) = N_z(t-1) + 1$ each time a new burst is transmitted on path π_z, with $N(0) = 0$. In the above expressions, the time index t refers to the time the source receives feedback from the network regarding the outcome (success or failure) of the most recent burst transmission along path π_z; similarly, index $t - 1$ refers to the time feedback was received regarding the immediately previous burst transmission over the same path. The priority of a path remains unchanged in the interval $[t-1,t)$. There are several ways for the source to receive feedback regarding a burst transmission. Without loss of generality, we assume that the feedback is provided by the signaling protocol, as is the case with the `Jumpstart` JIT signaling protocol [5]. Therefore, the update of path priorities is triggered by the reception of such feedback messages.

The priority of a path in expression (3.6) is simply the fraction of bursts that have been successfully transmitted along this path up to time t; hence, the range of path priorities is the interval (0,1). Therefore, at any given instant, the priority of a path is a measure of the likelihood that a burst transmission along this path will be successful, i.e., the burst will not be dropped due to contention at an intermediate OBS node.

At time t, the EPP strategy routes bursts from s to d along the path $\pi_{z^\star(t)}$ whose index $z^\star(t)$ is obtained using the following metric:

$$z^\star = \begin{cases} z, & prio(\pi_z,t) - prio(\pi_x,t) > \Delta \ \forall \ x \neq z \\ \arg\max_{1\leq z\leq m} \frac{prio(\pi_z,t)}{|\pi_z|}, & \text{otherwise} \end{cases} \tag{3.7}$$

In other words, if there is exists some path whose priority at time t is higher by the priority of all paths by an amount at least equal to a threshold Δ, then this path is selected for routing bursts. The threshold Δ reflects the degree of confidence in the selection of a given path for routing paths. If we are sufficiently confident that a path is better than others in terms of burst drop probability, then the selection is based solely on path priorities. Otherwise, we discount the priority of each path by its length, and we select a path based on the discounted priorities.

In our experiments, we have found that, as long as the traffic pattern does not change over time, the path priorities initially oscillate but eventually settle down (converge) to a certain value. Consequently, at steady-state, one of the candidate paths is always used for routing bursts between a given source-destination pair. Although we do not have a proof of convergence, we have observed such convergence for all path priorities over all the experiments we have conducted. This convergence procedure of the path priority values can be thought of as a "dynamic optimization" process: the path priority values for each source-destination pair keep affecting each other until a local minimum is reached. Numerical results to be presented later demonstrate that the paths selected in this manner perform better than shortest path routing in terms of burst drop probability.

3.3 Hybrid Path Switching Strategies

Each of the pure path switching strategies we described in the previous section uses some information regarding the network state in order to select one of a set of candidate paths for transmitting bursts between a source-destination pair. At the time a new burst is ready for transmission, the source uses a strategy to make a routing decision. We will say that a decision is "correct" if the outcome is a successful burst transmission, and "wrong" otherwise. In general, a strategy will be correct (i.e., make correct decisions) only some fraction of the time. Furthermore, at a given time and set of circumstances, different strategies may result in different decisions. Each pure path switching strategy uses only one piece of information in reaching a decision, and this information provides only a limited "view" of the network state.

In this section, we focus on hybrid strategies which, at each burst transmission instant, combine the decisions of several pure strategies into an overall decision in the hope of improving the accuracy of the path selection process and improve the overall burst drop probability. In general, a hybrid strategy emulates a set of pure strategies which run independently of each other "on the side." Each time a burst is ready to transmit, the decision of each pure strategy is computed, and the hybrid strategy uses a set of rules for selecting one of the decisions. The motivation for this approach is to combine the different partial "views" of the network state in a way that improves the performance. The next subsection presents a general framework for combining a set of pure strategies, followed by several instantiations of this framework into hybrid strategies.

3.3.1 A General Framework

The principles underlying the hybrid path switching strategies are based on ideas from the domain of machine learning [23, 45]. Specifically, it has been shown [45] that the *ensemble* decision reached by a set of voters is more accurate than the decision of any individual voter, provided that each voter reaches a decision in a manner that is largely independent of other voters. Consider, for instance, three voters, v_1, v_2, and v_3, who are called upon to make a binary decision. If the three voters use identical (or very similar) methods to reach a decision, then, whenever voter v_1 is wrong, voters v_2 and v_3 are likely to be wrong as well. However, if the errors made by the different voters are not correlated, then, whenever v_1 is wrong, it is possible that voters v_2 and v_3 are correct, so that a majority vote may reach the correct decision. Let us assume for simplicity that each voter has the same error rate $b < 1/2$ and that errors are independent. In this case, the probability that the majority vote will be wrong is equal to the area under the binomial distribution where more than one-half of the voters are wrong; this area diminishes quickly as the number of voters increases.

In the context of path switching in an OBS network, a pure path switching strategy corresponds to a voter, and the selection of a path corresponds to a (routing) decision. A strategy is "correct" if transmitting the burst over the path selected by the strategy is successful, and it is "wrong" if the burst is dropped along the path before it reaches its destination. We can think of the overall burst drop probability of a strategy as its "error rate," i.e., the fraction of time the method is incorrect in successfully selecting a path for a burst. Obviously, the drop probability *overestimates* the real error rate of the strategy, since the fact that a burst is dropped along a given path does not necessarily imply that the burst would have been successful had another path been chosen. Each of the pure strategies we have discussed exploits a different piece of information regarding the network state (e.g., bottleneck link congestion or utilization, or path priority) to reach a decision. Although these pieces of information partially overlap in the sense that they are based on observations of the state of network links, they are not identical, and provide separate views of the network. Therefore, we expect that making routing decisions by considering several different views simultaneously will lead to better performance in terms of burst drop probability.

In the remainder of this section, we consider a single source-destination pair (s, d). The source node s maintains $m > 1$ candidate paths, $\pi_1, \cdots, \pi_m$, for routing bursts to destination d. Therefore, for ease of presentation and to avoid repetition, we will drop any references to the source-destination pair (s, d). We also emphasize that our observations and hybrid path switching strategies apply similarly to all other source-destination pairs.

In order to formalize our approach, let us assume that there are n pure path switching strategies available, $S_1, S_2, \cdots, S_n$. A particular strategy S_i takes as input some information regarding the network state, and produces a probability distribution $p_i^{(z)}$ over the indices of the candidate paths; we shall discuss shortly how this probability distribution is obtained. The probability $p_i^{(z)}, z = 1, \cdots, m$, represents the degree of confidence that strategy S_i has in selecting candidate path π_z for routing the burst traffic. Obviously, we have: $p_i^{(1)} + p_i^{(2)} + \cdots + p_i^{(m)} = 1$.

A hybrid strategy H assigns a probability distribution q_i over the n pure path switching strategies $S_1, \cdots, S_n$. The probability q_i represents the degree of confidence of the hybrid strategy H that strategy S_i is correct in its selection of a path. Again, we have that: $q_1 + q_2 + \cdots + q_n = 1$. Then, the expected confidence of the hybrid strategy in selecting candidate path π_z is:

$$E_z = \sum_{i=1}^{n} q_i p_i^{(z)} \qquad z = 1, \cdots, m \tag{3.8}$$

Therefore, the decision of the hybrid strategy H is to route bursts along the path $\pi_z^\star$ with the maximum expected confidence, i.e., the one whose index $z^\star$ is given by:

$$z^\star = \arg \max_{1 \le z \le m} E_z \tag{3.9}$$

A hybrid strategy is characterized by the set of pure path switching strategies it utilizes and the probability distribution q_i it assigns over these strategies. In the following subsections, we introduce three hybrid strategies, each of increasing sophistication.

3.3.2 Majority Binary Voting (MBV) Strategy

Majority binary voting (MBV) is the simplest hybrid strategy. Let us assume that there are n pure strategies available, $S_i, \cdots, S_n$, where n is odd. Each strategy S_i makes a binary decision for each of the m candidate paths: whether to select it for routing bursts or not. Formally, the probability distribution $p_i^{(z)}$ returned by each strategy S_i is as follows:

$$p_i^{(z)} = \begin{cases} 1, & S_i \text{ selects path } \pi_z \\ 0, & \text{otherwise} \end{cases} \qquad i = 1, \cdots, n, \;\; z = 1, \cdots, m \tag{3.10}$$

The path selected by the hybrid MBV strategy is the one with the most number of votes. We note that this strategy assumes a uniform distribution q_i over the set of strategies $\{S_i\}$.

3.3.3 Weighted Non-Binary Voting (WNV) Strategy

The MBV strategy restricts the pure path switching strategies to vote for a single path, i.e., the one ranked the highest based on the metric used by the respective strategy. Non-binary voting allows each pure strategy S_i to assign a degree of confidence to each candidate path π_z through a probability distribution $p_i^{(z)}$. One straightforward approach to obtaining the probability distribution is to normalize the values $v_i^{(z)}$ (e.g., priority, congestion level, etc.) assigned to the various paths by strategy S_i:

$$p_i^{(z)} = \frac{v_i^{(z)}}{\sum_{l=1,\cdots,m} v_i^{(l)}} \qquad i = 1,\cdots,n, \; z = 1,\cdots,m \tag{3.11}$$

The weighted non-binary voting (WNV) strategy further assigns a probability distribution q_i over the set of pure strategies $\{S_i\}$, and reaches a decision using expressions (3.8) and (3.9). The main motivation for using a non-uniform distribution q_i is the fact that, as we shall demonstrate shortly, each pure strategy results in a different burst drop probability; furthermore, the relative performance of the various pure strategies depends on system parameters such as the network topology, the traffic load and pattern, etc. Since each strategy has a different "error rate," their contribution to the overall decision of the hybrid strategy should be weighted accordingly. In general, as we shall see, the performance of the hybrid strategy depends strongly on the choice of weights, with the best performance achieved when the weights reflect the relative error rate of the pure strategies.

3.3.4 Dynamic Weighted Non-Binary Voting (DWNV) Strategy

Under the WNV strategy, the probability distribution q_i over the set of pure strategies $\{S_i\}$ remains fixed at all times. One problem with such an approach is the difficulty in appropriately selecting the weights (degrees of confidence) q_i, since an inappropriate choice has the potential to result in poor performance. Instead, it would be desirable to have a method for dynamically adjusting the probability distribution q_i in real time (i.e., as the hybrid strategy is in operation) in a way that will minimize the overall burst drop probability; in this case, the probability distribution q_i would also converge to the optimal one. We now present a dynamic weighted non-binary voting (DWNV) strategy to achieve this objective.

Let $q(t) = (q_1(t),\cdots,q_n(t))$ be the probability distribution at time t, and let $B(t,q(t))$ be the burst drop probability of the hybrid strategy at time t when the current distribution is $q(t)$. Our objective is to obtain the distribution $q(t+1)$ at time $t+1$ such that the burst drop probability is minimized (recall that the time indices refer to the times a burst is ready to be transmitted between the given source-destination pair). In other words, we need to select the distribution $q^\star(t+1)$ such that:

$$q^\star(t+1) \quad = \quad \arg \min_{q(t+1)} B(t+1, q(t+1)) \tag{3.12}$$

However, it is not possible to solve the above optimization problem directly. We therefore employ a heuristic to dynamically update the q-distribution. We assume that the confidence $c_i(t)$ in the decision of a strategy S_i is reversely proportional to its burst drop probability $b_i(t)$ at time t:

$$c_i(t) \quad = \quad \frac{1}{b_i(t)+\epsilon} \qquad i = 1, \cdots, n \tag{3.13}$$

where ϵ is a smoothing value to avoid division by zero when $b_i = 0$. Based on the confidence c_i of choosing strategy S_i, we compute the new weight q_i as:

$$q_i(t+1) \quad = \quad \frac{c_i(t)}{\sum_{l=1,\cdots,n} c_l(t)} \qquad i = 1, \cdots, n \tag{3.14}$$

The computation of each of the expressions (3.14) warrants further discussion. The overall burst drop probability $B(t, q(t))$ of the hybrid policy is calculated at the source node using the feedback messages from the network. However, it is not possible for the source node to calculate directly (i.e., based only on feedback from the network) the burst drop probability $b_i(t)$ of each pure strategy S_i as required by expression (3.14). To see why a direct calculation of $b_i(t)$ is not possible, consider what happens if the hybrid strategy adopts a decision that is different than the decision of some pure strategy S_i. In this case, the feedback received by the source provides information regarding the decision made by the hybrid policy but no information regarding the decision made by pure strategy S_i; in other words, the source has no way of knowing with certainty whether the burst transmission would have been successful had it used the path selected by S_i instead.

To overcome this difficulty, we use the following approach to compute the burst drop probability $b_i(t)$ for a pure strategy S_i whose decision at time t does not coincide with the decision of the hybrid strategy. Let π be the path chosen by S_i, and let $prio(\pi, t)$ be the priority of (burst drop probability along) this path; this priority is computed in the course of the operation of the hybrid policy as in expression (3.6). Then, we use $prio(\pi_t)$ to update the drop probability of strategy S_i, by making the approximation that the outcome of routing a burst over path π at time t will be failure with probability $1 - prio(\pi, t)$ and success with probability $prio(\pi, t)$. Of course, for any pure strategy whose decision coincides with that of the hybrid, the drop probability can be directly updated based on feedback from the network. Therefore, the burst drop probability for any pure strategy S_i whose decision at time t is to use path π, is updated as follows:

$$b_i(t+1) \quad = \quad \begin{cases} 0, & t = 0 \\ \frac{b_i(t)\times N}{N+1}, & \text{hybrid strategy chose path } \pi \text{ and burst was successful} \\ \frac{b_i(t)\times N+1}{N+1}, & \text{hybrid strategy chose path } \pi \text{ and burst was dropped} \\ \frac{b_i(t)\times N+(1-prio(\pi,t))}{N+1}, & \text{hybrid strategy did not choose path } \pi \end{cases} \tag{3.15}$$

3.4 Numerical Results

In this section, we use simulation to investigate the performance benefits of path switching in OBS networks. We use the method of batch means to estimate the burst drop probability, with each simulation run lasting until 6×10^5 bursts have been transmitted in the entire network. We have also obtained 95% confidence intervals for all our results; however, they are so narrow that we omit them from the figures we present in this section in order to improve readability.

We consider two different traffic patterns in our study:

- **Uniform pattern:** each switch generates the same traffic load, and the traffic from a given switch is uniformly distributed to other switches.

- **Distance-dependent pattern:** the traffic between a pair of switches is proportional to $|\pi|$ (if $|\pi| = 1$), or $\frac{|\pi|}{|\pi|-1}$ (if $|\pi| > 1$); here, π denotes the shortest path between the pair of switches.

We used two 16-node networks in our simulation experiments. The 4×4 torus network shown in Figure 3.1 is based on a regular topology, while the network in Figure 3.2 is based on an irregular topology derived from the 14-node NSF network. All the figures in this section plot the burst drop probability against the "normalized network load" ρ_W, which is obtained by dividing the total load offered to the network by the number W of wavelengths: $\rho_W = \dfrac{\sum_{ij} \rho_{ij}}{W}$.

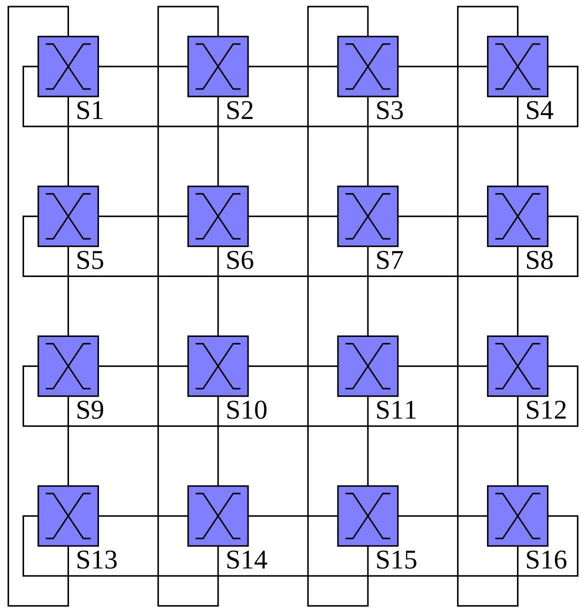

Figure 3.1: The 4 × 4 torus network

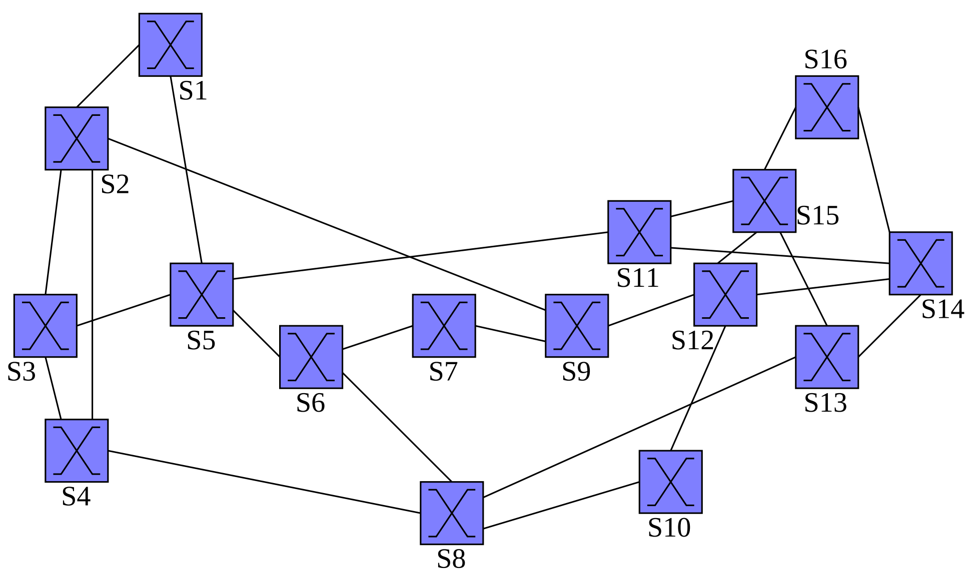

Figure 3.2: The 16-node topology based on the 14-node NSFNet

3.4.1 Pure Path Switching Strategies

Let us first investigate the performance improvement that is possible with path switching over shortest path routing. Shortest path routing is widely used in both circuit-switched and packet-switched networks since it minimizes the delay and optimizes the utilization of resources. However, shortest path routing does not take into consideration the traffic load offered to the network, and it often causes certain links to become congested while other links (which happen to lie along longer paths) remain underutilized. Such a scenario is highly undesirable in OBS networks in which burst drop probability is the primary performance metric of interest: since it is generally assumed that intermediate switches do not buffer bursts, having a few highly congested links may lead to unacceptably high burst loss for the entire network. Path switching may help alleviate this problem in a distributed manner by having each source select an appropriate path to each destination; the sources do not interact explicitly with each other, but only implicitly through the feedback they receive from the network. In our experiments, we assume that each source has to select among $m = 2$ candidate paths to each destination; these are the two shortest link-disjoint paths in the network for the given source-destination pair.

We now compare four routing schemes:

- **Shortest-path (SP) routing:** bursts are routed over the shortest path (in terms of hops) between source and destination, with ties broken arbitrarily.
- **WBLU path switching:** bursts are routed over the path determined by the WBLU path switching strategy.
- **WLC path switching:** bursts are sent over the path determined by the WLC strategy.
- **EPP path switching:** bursts are routed over the path determined by the EPP strategy.

Figures 3.3-3.4 plot the burst drop probability of the above four routing schemes for the NSF network with uniform traffic. Figure 3.3 (respectively, Figure 3.4) plots the burst drop probability for low (respectively, high) loads. As we can see, all three path switching strategies perform consistently better than SP routing throughout the load range considered in the figures; the only exception is at very high loads, where the high burst drop probability is due to a saturated network. This result demonstrates the benefits of path switching over shortest path routing.

Another important observation from the two figures is that none of the three path switching strategies is a clear winner over the entire range of loads shown. In general, WBLU performs the best at low loads, EPP is the best strategy at high loads, while the burst drop probability of WLC is between the values of the other two strategies. Note that, at low network loads, most links have low utilization, and avoiding the few highly utilized (bottleneck) links can significantly improve the burst drop probability. Since the WBLU strategy takes account the bottleneck link utilization in determining the burst path, it is not surprising that it performs well at low loads. At high loads, on the other hand, the EPP strategy outperforms the WBLU and WLC strategies. We believe that this behavior can be explained by the manner in which the three strategies update their path decisions. Under EPP, path priorities are updated immediately upon the receipt of feedback messages from the network, whereas the WBLU and WLC strategies update their routing decisions periodically (i.e., once they receive the most recent information on link utilization or congestion, which in turn is updated periodically). The period of update for WBLU and WLC is independent of the network load. With the EPP strategy, however, as the load (i.e., the rate of transmitted bursts) increases, the rate of feedback from the network increases accordingly, providing a more accurate view of the network state and resulting in better routing decisions.

The performance of the four routing methods for the Torus network and uniform traffic is shown in Figures 3.5-3.6. The WLC and EPP strategies perform consistently better than SP routing, and in fact the burst drop probability of EPP is significantly lower than that of both WLC and SP across the whole range from low to high loads. The WBLU strategy, on the other hand, is only slightly better than SP at low loads, and slightly worse than SP at high loads. This result can be explained by the fact that WBLU makes a routing decision based only on the utilization of the bottleneck link. In a symmetric topology such as the Torus, the WBLU strategy leads to routing oscillations which tend to hurt the overall performance. In our experiments, we have observed that the oscillations persist throughout the simulation, and that they become worse as the offered load increases. In the asymmetric NSF network, on the other hand, we have observed that the routing decision of WBLU oscillates at first, but it later settles down to a fixed path. The only exception is at very high loads when the bottleneck links are saturated, in which case WBLU keeps

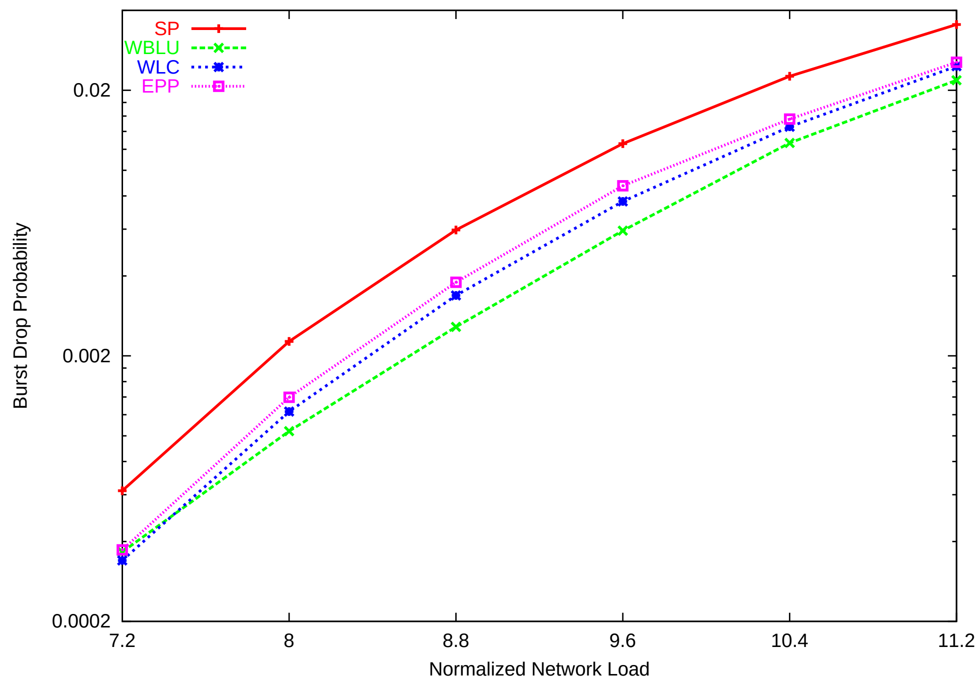

Figure 3.3: Burst drop probability for NSF network, uniform traffic, Low load

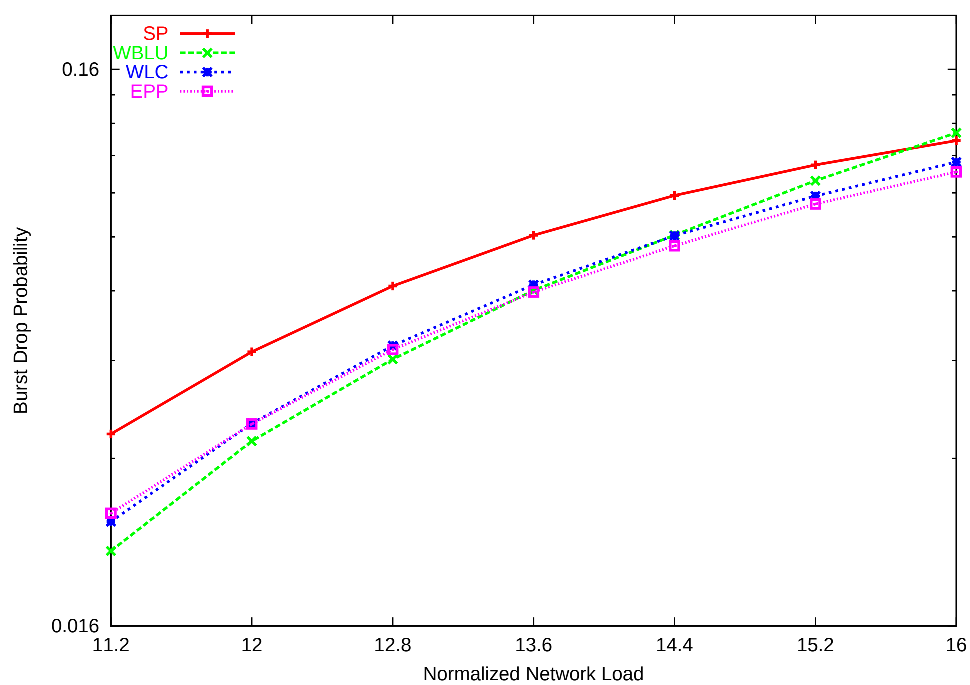

Figure 3.4: Burst drop probability for NSF network, uniform traffic, High load

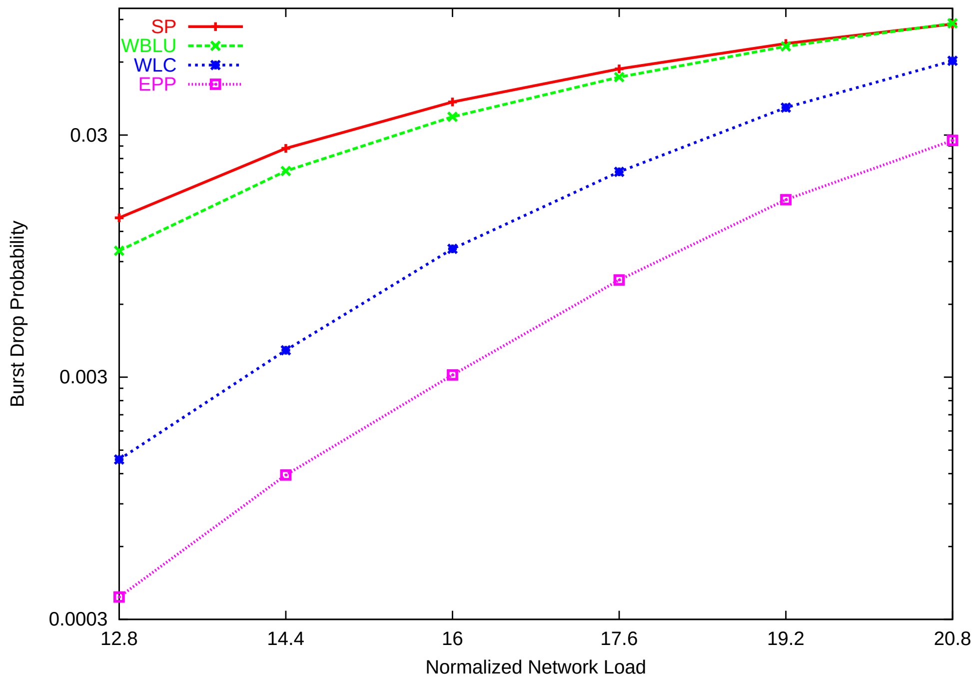

Figure 3.5: Burst drop probability for Torus network, uniform traffic, Low load

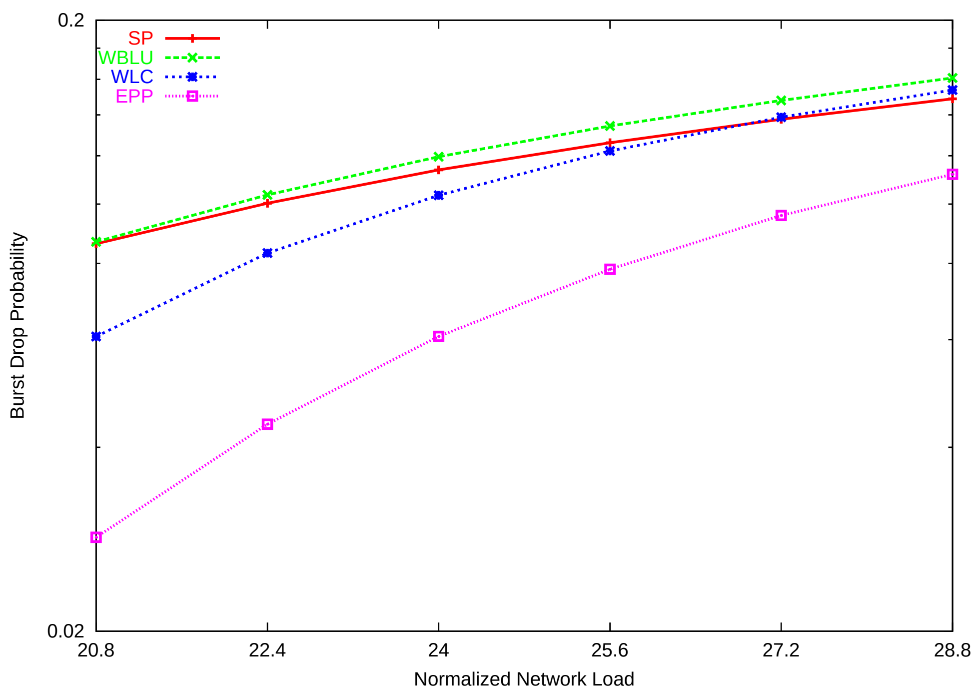

Figure 3.6: Burst drop probability for Torus network, uniform traffic, High load

oscillating among the candidate paths; this is reflected in Figure 3.4 for a load of 16, when WBLU performs worse than SP routing.

Finally, Figures 3.7-3.8 and 3.9-3.10 show the burst drop probability of the four routing methods for the NSF and Torus networks, respectively, with distance-dependent traffic. As we can see, the relative performance of the four routing methods is similar to that under uniform traffic.

Based on the results presented in this section, we conclude that an appropriately selected path switching strategy can significantly reduce the burst drop probability over SP routing. However, the relative performance of a given path switching strategy depends on the network topology and load, and is difficult to characterize *a priori*. Therefore, in the next section we investigate hybrid path switching strategies which combine several independent pure path switching methods so as to provide consistently good performance.

3.4.2 Hybrid Path Switching Strategies

We now consider the WNV and DWNV hybrid path switching strategies we introduced in Chapter 3.3. Each hybrid strategy utilizes four routing strategies in making its decision: SP routing and the WBLU, WLC, and EPP pure path switching strategies. In order to characterize the performance of hybrid path switching, in this section we compare the following three routing schemes:

- **WNV path switching:** bursts are sent along the path with the maximum weighted overall confidence. By definition, WNV assigns static weights (q-distributions) to each of the four pure strategies (voters). We have found that different weights perform differently for each of the two topologies we consider here. Therefore, after some experimentation, we have used the following weights: for the NSF network, all weights are equal to 1/4 (a uniform distribution), while for the Torus network, the weights are: 1/8 (SP and WBLU), 1/4 (WLC), and 1/2 (EPP).

- **DWNV path switching:** bursts are sent along the path with the maximum weighted overall confidence. Under this strategy, initially the weights of all pure strategies are equal, but they are adjusted dynamically during the operation of the network as we explained in Chapter 3.3.

- **Best pure strategy:** bursts are sent along the path determined by the pure strategy

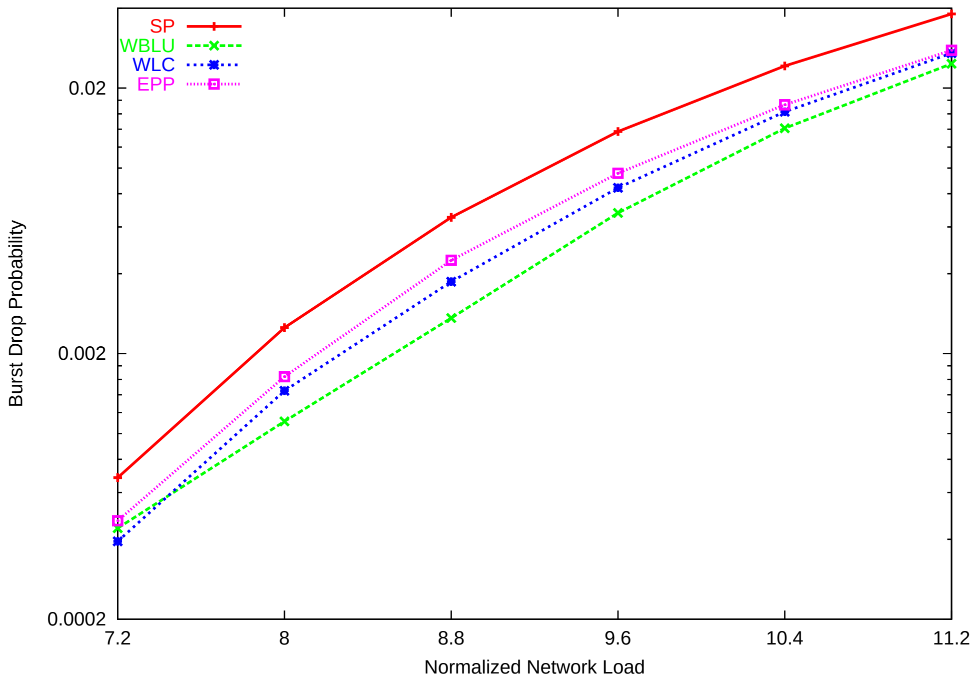

Figure 3.7: Burst drop probability for NSF network, distance-dependent traffic, Low load

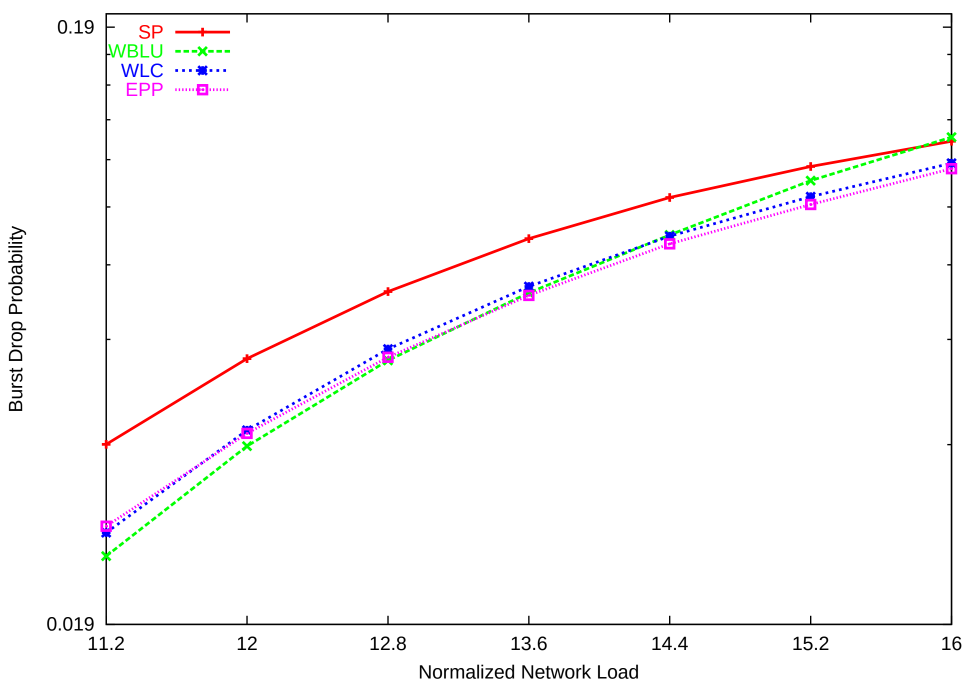

Figure 3.8: Burst drop probability for NSF network, distance-dependent traffic, High load

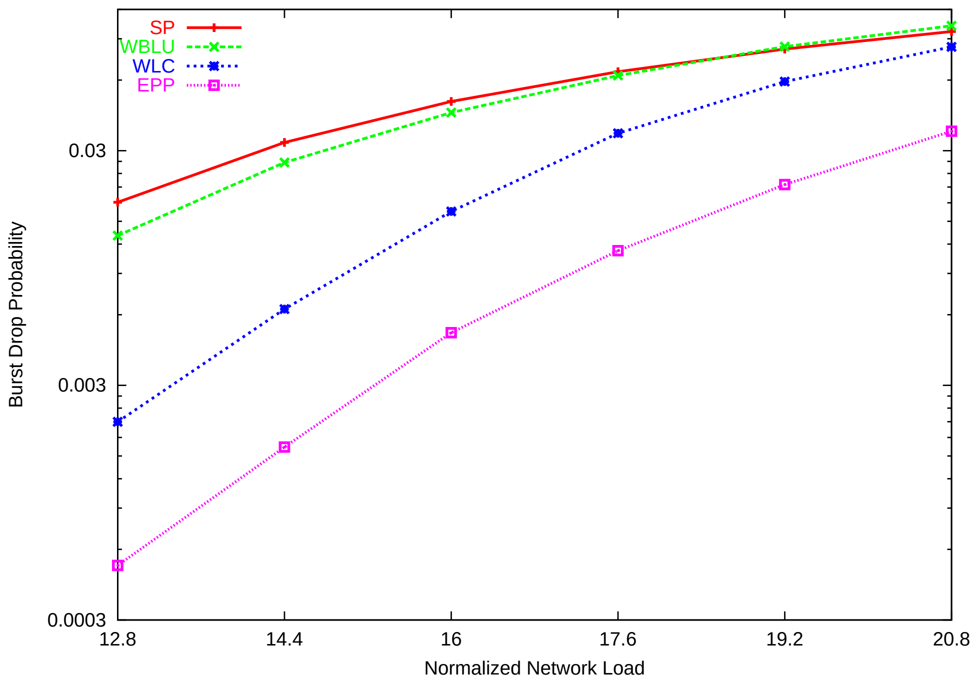

Figure 3.9: Burst drop probability for Torus network, distance-dependent traffic, Low load

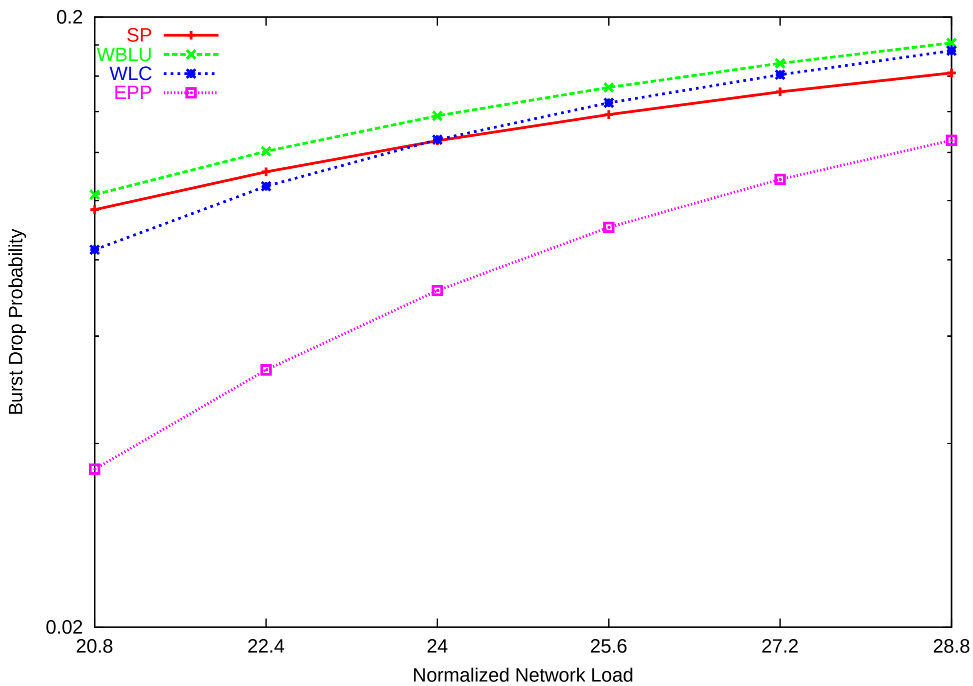

Figure 3.10: Burst drop probability for Torus network, distance-dependent traffic, High load

with the best performance among the four strategies SP, WBLU, WLC, and EPP. Note that, if it turns out that if one pure strategy is best across some range of loads while another strategy is best across a different range, we will present both strategies.

Figures 3.11-3.12 compare DWNV, WNV and the two best pure path switching strategies for the NSF network with uniform traffic; note that WBLU has the best performance among the pure path switching strategies at low loads, while EPP is the best pure strategy at high loads, consistent with the behavior we observed in the previous subsection. We also observe that the hybrid WNV path switching scheme improves the burst drop probability over both pure strategies; in effect, the WNV curve tracks the best of the WBLU or EPP curves. This result confirms our intuition that combining and taking into account several different views of the network state increases the performance. We also note that when the weight of each pure strategy is adjusted dynamically to reflect the real-time network performance, as accomplished by the DWNV hybrid strategy, the burst drop probability is further improved. Our experiments demonstrate that through dynamic adjustments, the weights assigned to each pure strategy by a source-destination pair are tuned to prefer one (or the combination of more than one) pure strategy. Which strategies are preferred depends on the source-destination pair (i.e., path through the network) and the traffic pattern. This tuning procedure can be viewed as a "dynamic optimization" process that allows the hybrid DWNV strategy to achieve a final set of weights (q-distribution) that is near-optimal in the sense of minimizing the burst drop probability.

Similar observations to the above can be made from Figures 3.13-3.14 which present results for the NSF network with the distance-dependent traffic pattern.

Figures 3.15-3.16 and 3.17-3.18 present results for the Torus network with the uniform and distance-dependent traffic patterns, respectively. The figures compare the performance of the WNV and DWNV hybrid strategies and the EPP strategy, which is the best performing pure strategy for the Torus. The hybrid strategies perform slightly better than EPP, except at low loads. The fact that EPP outperforms the hybrid strategies at low loads can be explained by referring to Figures 3.5-3.6 and 3.9-3.10. As we can see, EPP performs significantly better than the other pure strategies at low loads. Recall that the hybrid strategies take into account the decisions of the other pure strategies (in addition to EPP), which tend to be wrong most of the time. Therefore, the hybrid strategies will tend to perform slightly worse than EPP at the low loads. However, at moderate and high loads

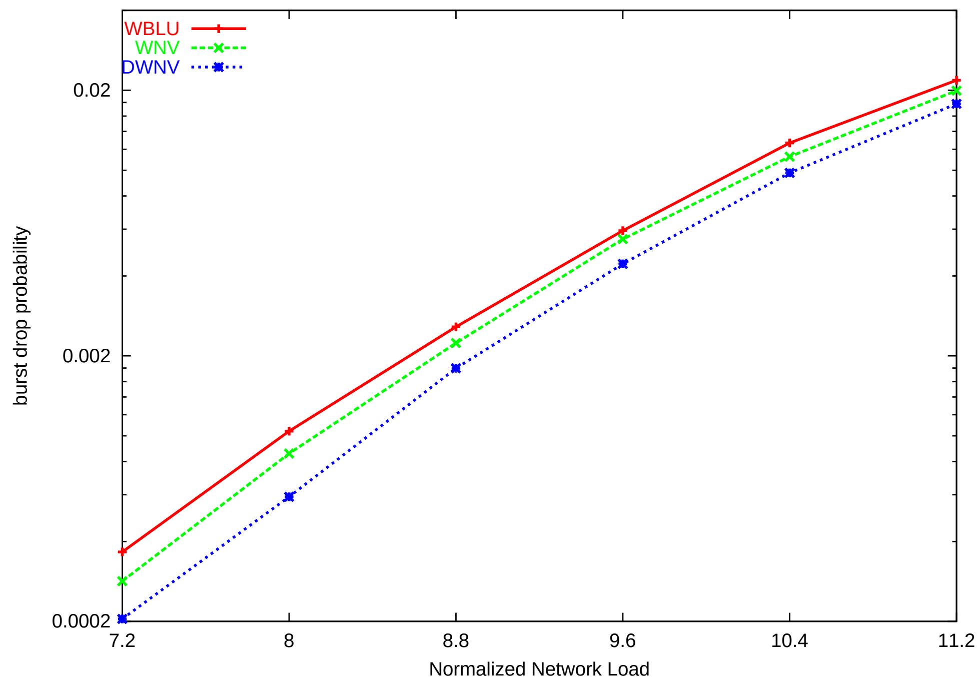

Figure 3.11: Burst drop probability for NSF network, uniform traffic, Low load

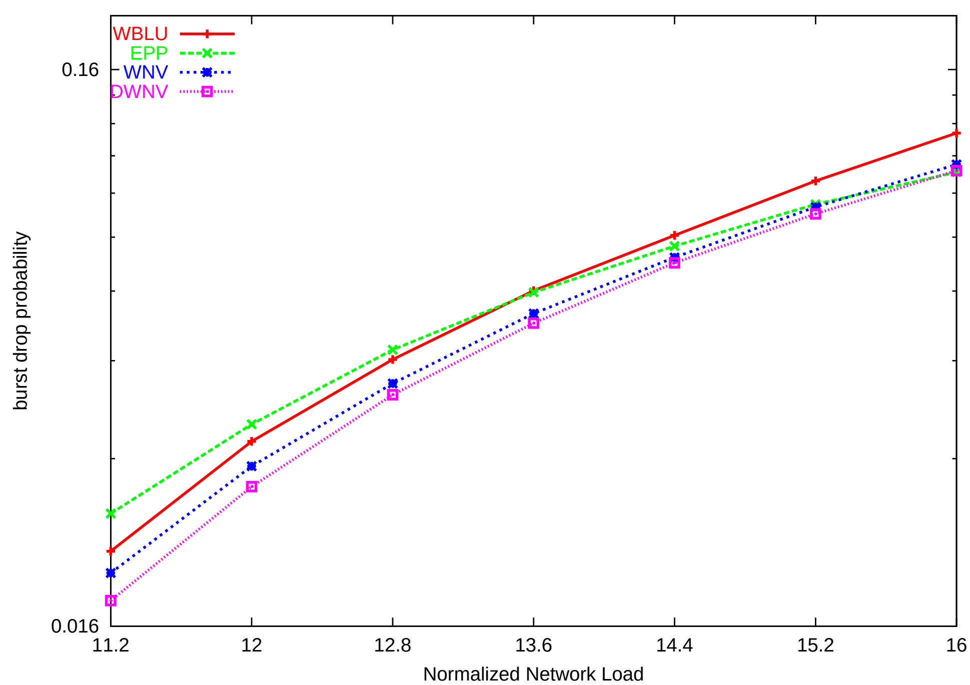

Figure 3.12: Burst drop probability for NSF network, uniform traffic, High load

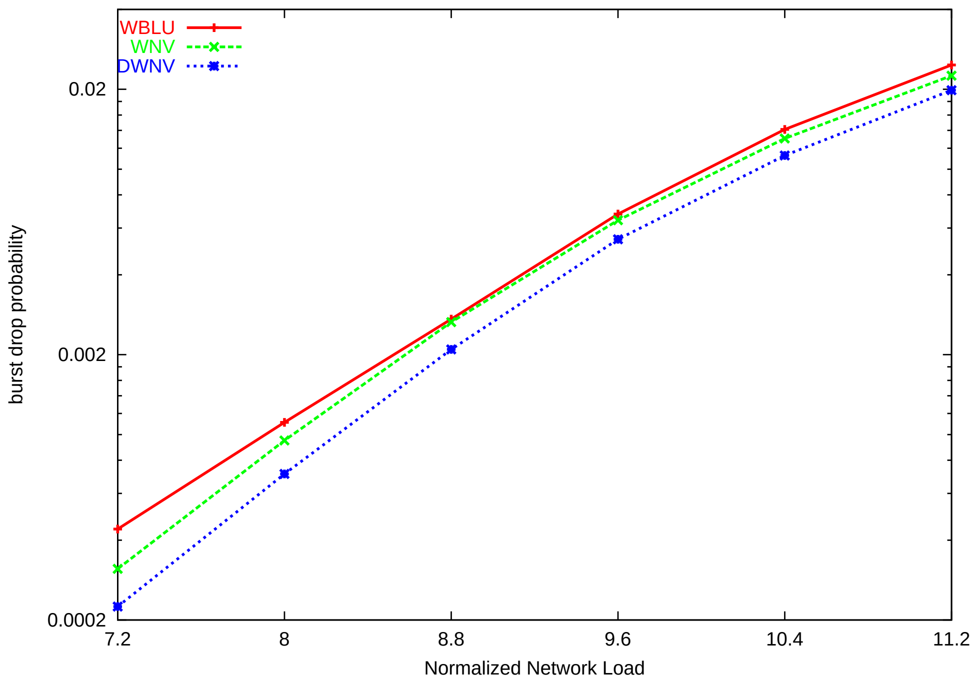

Figure 3.13: Burst drop probability for NSF network, distance-dependent traffic, Low load

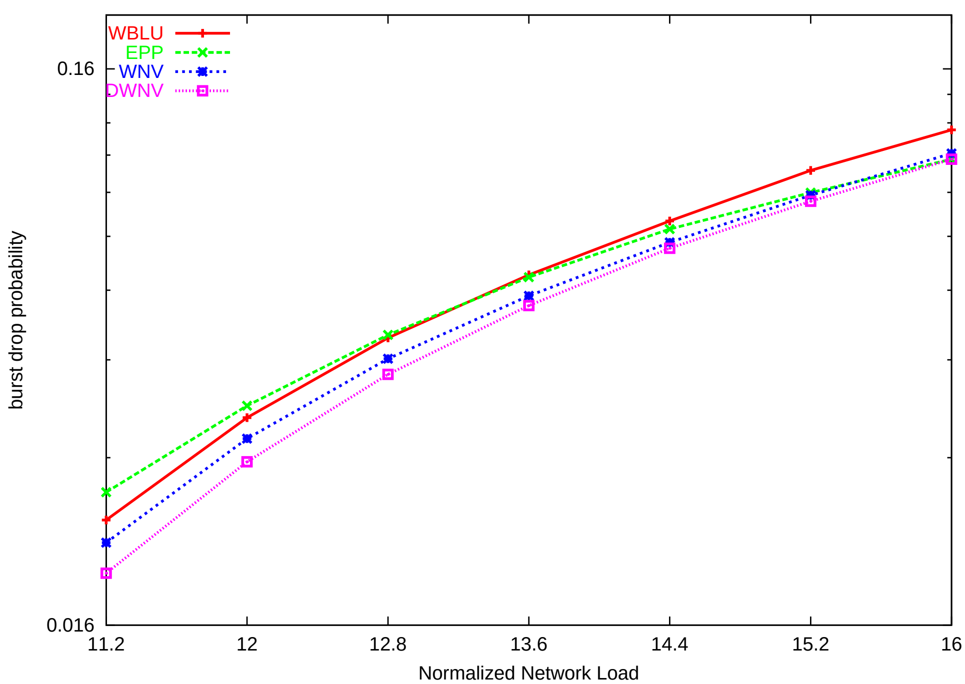

Figure 3.14: Burst drop probability for NSF network, distance-dependent traffic, High load

when the performance difference between EPP and the other pure strategies diminishes, the hybrid strategies are successful in improving the performance compared to EPP.

Another observation from Figures 3.15-3.16 and 3.17-3.18 is that WNV sometimes performs better than DWNV. This result is due to the fact that we have experimented with a large set of different static weights for the WNV strategy, and we have shown the result for the best one here. Unfortunately, selecting the set of static weights to optimize the performance is a difficult, time-consuming task, that requires good understanding of the relative performance of the pure strategies for a given topology, traffic pattern, and network load. When it is not practical, feasible, or possible to optimize the static weights for WNV, then the DWNV strategy that dynamically adjusts the weights is the best choice.

Overall, we can summarize our results as follows.

- Pure path switching strategies can reduce the burst drop probability in an OBS network compared to shortest path routing.

- The performance improvement depends on various parameters, including the congestion information utilized by the path switching strategy, the network topology, the traffic pattern, and the network load; in many cases, the performance improvement over shortest path routing can be dramatic.

- Hybrid path switching strategies can be used to further improve the network performance. However, if there is a single pure switching strategy that clearly outperforms all others for a given set of parameters within the network's operating region, then a hybrid strategy may not provide any improvement (since it relies on strategies other than the best one). In this case, it is best to simply use the most successful pure strategy instead.

- If it is possible to appropriately select the weights assigned to the pure strategies by a hybrid strategy, then the overall network performance is optimized. Otherwise, a hybrid strategy that dynamically adjusts the weights performs best.

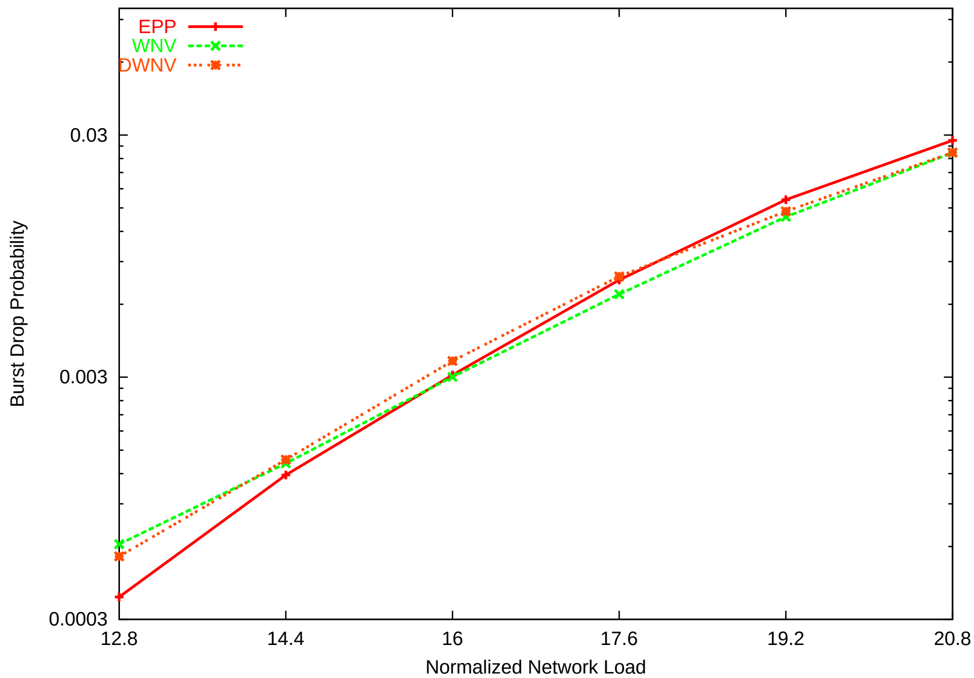

Figure 3.15: Burst drop probability for Torus network, uniform traffic, Low load

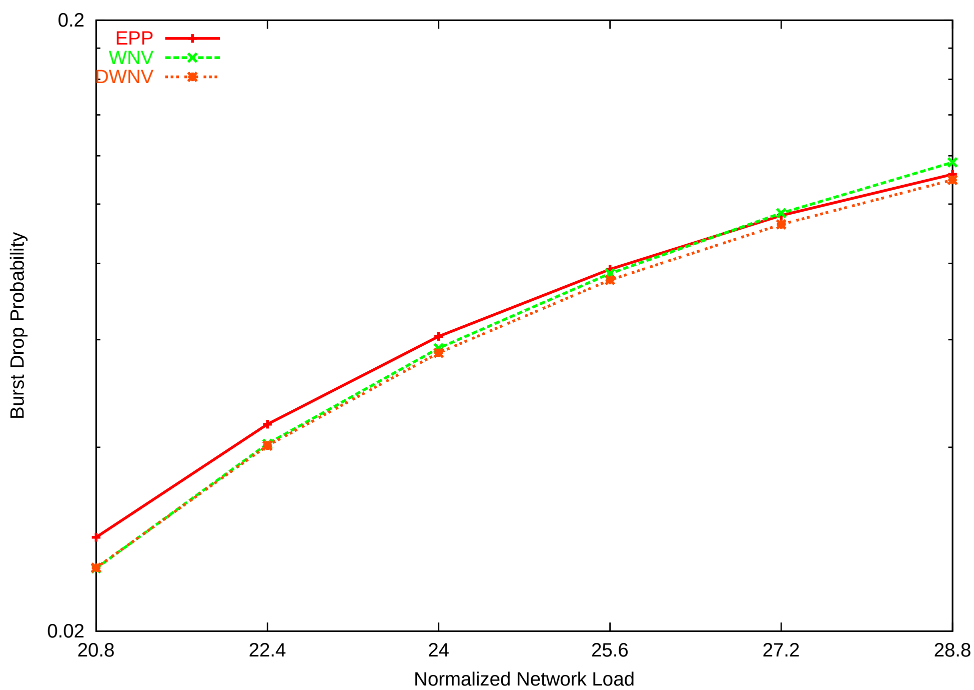

Figure 3.16: Burst drop probability for Torus network, uniform traffic, High load

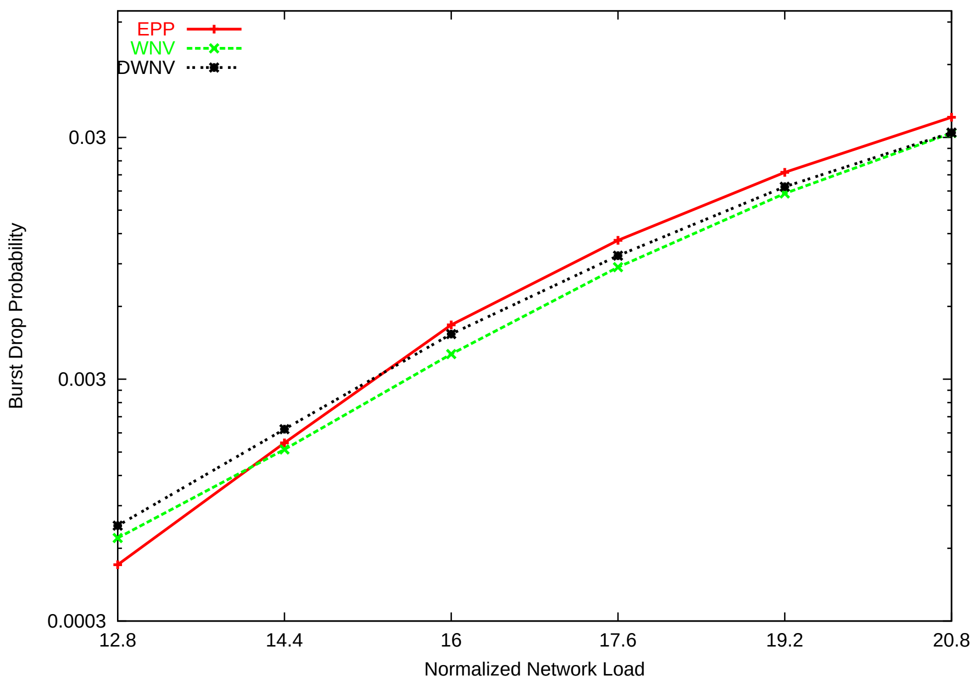

Figure 3.17: Burst drop probability for Torus network, distance-dependent traffic, Low load

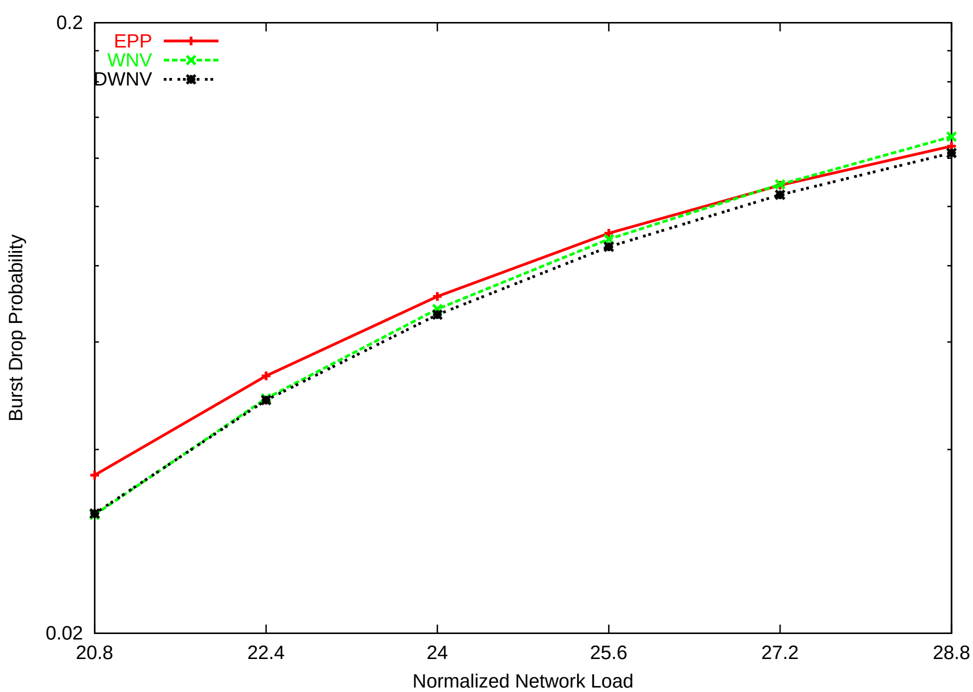

Figure 3.18: Burst drop probability for Torus network, distance-dependent traffic, High load

Chapter 4

A Framework for Absolute QoS Guarantees in OBS Networks

In this chapter we develop a general framework for absolute service guarantees to users of an OBS network in terms of the end-to-end burst loss. We first present a parameterized model for wavelength sharing. Then, we develop a heuristic for optimizing the policy parameters to support per-link absolute QoS guarantees. Finally, we present a methodology for translating the end-to-end QoS requirements into appropriate per-link parameters so as to provide network-wide guarantees. Our approach is easy to implement, it can support a wide variety of traffic classes, and is effective meeting the QoS requirements and keeping the loss rate of best-effort traffic low.

This chapter is organized as follows. In Chapter 4.1, we discuss the assumptions regarding the OBS network we consider in this study. In Chapter 4.2 we present a suite of parameterized wavelength sharing policies, and in Chapter 4.3 we develop an algorithm for optimizing the policy parameters for a single OBS link. In Chapter 4.4 we extend our model to an OBS network and introduce an algorithm for determining near-optimal link policy parameters from the end-to-end QoS requirements, traffic statistics, and network properties. We present numerical results to validate our approach in Chapter 4.5.

4.1 The OBS Network Under Study

Our assumptions regarding the OBS network are similar to those in Chapter 3.1. We consider an OBS network with N nodes. Each link in the network can carry burst traffic on any wavelength from a fixed set of W wavelengths, $\{\lambda_1, \lambda_2, \cdots, \lambda_W\}$. We assume that each OBS node is capable of full wavelength conversion, hence an incoming burst can be forwarded on any wavelength available at its output port regardless of the wavelength on which it arrived. The network does not use any other contention resolution mechanism. Specifically, OBS nodes do not employ any buffering, either electronic or optical, in the data path, and they do not utilize deflection routing or burst segmentation. Therefore, if a burst requires an output port at a time when all wavelengths of that port are busy transmitting other bursts, then the burst is dropped.

The network supports P classes of traffic, where P is a small integer. Once assembled at the edge of the network, a burst is assigned to one of the P classes; the mechanism for assigning bursts to traffic classes is outside the scope of our work. The class to which a burst belongs is recorded in the SETUP (control) message that precedes the burst transmission. We assume that intermediate nodes make forwarding decisions by taking into account both the availability of resources (e.g., the number of free wavelengths at an output port) and the information regarding the class of a burst. Specifically, an intermediate node may drop a burst of a lower priority class even when there are wavelengths available at its outgoing link. In the next section, we describe a set of policies that intermediate nodes follow when forwarding bursts.

Each traffic class $i, i = 1, \cdots, P-1$, is characterized by a worst-case *end-to-end* loss guarantee B_i^{e2e}. Parameter B_i^{e2e} represents the long-run fraction of bursts from class i that are dropped by the network before reaching their destination. Without loss of generality, we assume that bursts of class i have more stringent loss requirements than bursts of class j, when $i < j$; in other words:

$$B_i^{e2e} \quad < \quad B_j^{e2e}, \qquad 1 \leq i < j \leq P \tag{4.1}$$

Bursts of class P are not associated with any worst-case loss guarantee; consequently, we will refer to class P as the *best-effort* class, and, for convenience, we will set $B_P^{e2e} = 1.0$.

The objective of the network provider, and the one we consider in this work, is to:

> *ensure that the loss rate of class $i, i = 1, \cdots, P-1$, does not exceed its worst-case loss guarantee B_i^{e2e}, while at the same time minimizing the loss rate of the*

best-effort class P.

In order to achieve this objective, the network nodes need to employ appropriately designed mechanisms to allocate wavelength resources to bursts of each class based on its load and worst-case loss requirement. In the following, we develop a suite of wavelength sharing policies and evaluate their performance.

4.2 Wavelength Sharing Policies: The Single Link Case

In this section we consider a single link of an OBS network, and we present a set of policies to support different classes of traffic sharing the wavelength resources of the link. The techniques we propose allow for (limited) resource sharing among classes, but also offer each class varying degrees of protection from other classes. The ideas underlying our policies arise naturally in practice, and have been considered before: in the specific setting of memory allocation in network nodes [54], and in the more general context of resource sharing [18]. Our main contribution in this section is to develop analytical methods to calculate the burst loss probability for the various traffic classes under each policy. The analytical methods are the first step towards the design of effective mechanisms to provide absolute end-to-end QoS guarantees in OBS networks, a task we undertake in the following two sections.

We assume that the (unidirectional) OBS link under study consists of W parallel wavelengths, and carries P classes of bursts. The policies we consider manage the wavelength space by associating with each traffic class a pair of values that impose bounds on the use of the link's transmission resources by the class:

- W_i^{max}, referred to as *wavelength upper bound for class i*, is the maximum number of wavelengths that may be occupied simultaneously by bursts of class i. Setting $W_i^{max} < W$ ensures that class i bursts will not consume all available wavelengths at any given time, thus providing a form of protection to other traffic classes from class i.

- W_i^{min}, referred to as *wavelength lower bound for class i*, is the minimum number of wavelengths set aside (reserved) by the link for class i bursts. Whenever $W_i^{min} > 0$, the lower bound guarantees that there is always space for a specified number of bursts from class i, in essence protecting this class in case other classes experience (transient or permanent) overload.

By specifying values for the pair of bounds (W_i^{min}, W_i^{max}) for each traffic class i, a policy may strike any desired balance between two conflicting objectives: *QoS protection*, through class separation, and *efficient utilization*, through sharing of wavelength resources.

We note that a *complete wavelength sharing* policy dictates that:

$$W_i^{min} = 0, \qquad W_i^{max} = W, \qquad i = 1, \cdots, P \tag{4.2}$$

Such a policy offers no protection, and cannot provide any differentiation among bursts with respect to loss guarantees. Therefore, we do not consider this policy in our work.

In the following subsections, we present four broad classes of policies as determined by the range of values that the lower and upper bounds, W_i^{min} and W_i^{max}, respectively, are allowed to take. We also present analytical models for computing the burst loss probability for each policy, assuming that the pair of values (W_i^{min}, W_i^{max}) for each class i are known in advance; how to determine these values so as to achieve the objective stated in Chapter 4.1 is the subject of Chapter 4.3. The analytical models are derived based on the assumption that traffic class $i, i = 1, \cdots, P$, is characterized by a Poisson arrival rate λ_i, and mean holding time μ_i. We also let $\rho_i = \lambda_i / \mu_i$ denote the offered load of class i to the link.

4.2.1 Wavelength Partitioning (WP)

The wavelength partitioning (WP) policy partitions the wavelength space such that each of the P traffic classes has dedicated access to a subset of the W wavelengths. More specifically, the wavelength bounds for the traffic classes are defined as:

$$0 < W_i^{min} = W_i^{max} = W_i < W, \quad i = 1, \cdots, P \tag{4.3}$$

with the additional constraint that the sum of the number of wavelength dedicated to each class must equal to the number of available wavelengths:

$$\sum_{i=1}^{P} W_i = W \tag{4.4}$$

More specifically, bursts arriving at a link following the WP policy are handled as follows:

when a class-i burst arrives, if the number n_i of wavelengths busy with class-i bursts is less than W_i, the burst is transmitted on any free wavelength; otherwise, it is simply dropped.

Clearly, the WP policy and the complete sharing policy defined by expression (4.2) are at the opposite ends of the spectrum of possible wavelength sharing policies.

The WP policy was considered earlier in the context of OBS networks in [113], where it was referred to as dynamic wavelength grouping (DWG). We adopt it here as a baseline policy against which to compare the policies we present next. A link using the WP policy operates as P independent $M/M/m/m$ queueing systems, one per traffic class. The drop probability B_i for class-i bursts can be computed using the well-known Erlang-B formula for an $M/M/m/m$ system:

$$B_i = \frac{{\rho_i}^{W_i}/W_i!}{\sum_{j=0}^{W_i} {\rho_i}^j/j!} \tag{4.5}$$

WP is easy to implement, as at any time t, one only needs to keep track of the number of wavelengths occupied by bursts of each class. Its main drawback is the lack of statistical multiplexing of bursts from different classes, which can lead to a substantial increase in the number of wavelengths required to guarantee a given level of QoS for each class. As suggested in [54], the performance of complete partitioning can be improved if some sharing of resources is introduced. In the next three subsections we describe policies which provide different levels of wavelength sharing among the various traffic classes.

4.2.2 Wavelength Sharing with Maximum Occupancy (WS-Max)

In this scheme, we make all classes share the whole wavelength space, but we restrict the level of sharing by imposing an upper bound, $W_i^{max} < W$, on the number of wavelengths that class i can use at any given time. On the other hand, the wavelength lower bound for each class is set to zero:

$$0 = W_i^{min} < W_i^{max} < W, \quad i = 1, \cdots, P \tag{4.6}$$

To allow for wavelength sharing, the sum of the wavelength upper bounds over all traffic classes must exceed the number of available wavelengths, i.e.,

$$\sum_{i=1}^{P} W_i^{max} > W \tag{4.7}$$

More formally, the WS-Max policy operates as follows:

> *when a class-i burst arrives, if the number n_i of wavelengths busy with class-i bursts is less than W_i^{max}, the burst is transmitted on any free wavelength if one exists; otherwise the burst is dropped.*

To obtain the burst drop probability under the WS-Max policy, we observe that the state of the OBS link can be described by the vector $\underline{n} = (n_1, \cdots, n_P)$, where n_i is a nonnegative random variable denoting the number of class-i bursts. The Markovian process describing the evolution of the OBS link is a truncated process of the P independent $M/M/m/m$ queues. Specifically, the set of feasible states S of the truncated process can be described as:

$$S = \left\{ \underline{n} \mid \sum_{i=1}^{P} n_i \leq W, \; n_i \leq W_i^{max}, \; i = 1, \cdots, P \right\} \tag{4.8}$$

Then, the steady state probability of the truncated process has the following product form [57]:

$$P(\underline{n}) = P(n_1, n_2, \cdots, n_P) = \begin{cases} C \, \frac{\rho_1^{n_1}}{n_1!} \frac{\rho_2^{n_2}}{n_2!} \cdots \frac{\rho_P^{n_P}}{n_P!}, & \underline{n} \in S \\ 0, & \underline{n} \notin S \end{cases} \tag{4.9}$$

where C is the normalizing constant representing the probability that the OBS link is idle (that is, $C = P(\underline{0})$).

The normalizing constant C can be computed as:

$$C^{-1} = \sum_{\underline{n} \in S} \frac{\rho_1^{n_1}}{n_1!} \frac{\rho_2^{n_2}}{n_2!} \cdots \frac{\rho_P^{n_P}}{n_P!} \tag{4.10}$$

Since the cardinality of the state space S is almost $O(W^P)$ [56], a brute force calculation of C^{-1} is computationally expensive for links with a large number of wavelengths and/or traffic classes.

An effective algorithm for calculating the normalizing constant (and, consequently, the steady-state blocking probabilities) for a class of resource-sharing models was proposed in [16]. This algorithm is based on the numerical inversion approach introduced in [17]. In this work, we adopt the direct method in [16], which is appropriate for the system sizes we consider, and we calculate the normalizing constant via the P-fold nested sum:

$$C^{-1} = \sum_{n_1=0}^{K_1} \frac{\rho_1^{n_1}}{n_1!} \sum_{n_2=0}^{K_2} \frac{\rho_2^{n_2}}{n_2!} \cdots \sum_{n_P=0}^{K_P} \frac{\rho_P^{n_P}}{n_P!} \tag{4.11}$$

The P upper limits $K_i,\ i = 1, \cdots, P$, in the above summation are not independent; their value depends on both the wavelength upper bound W_i^{max} and the values of parameters $n_k, k = 1, \cdots, i-1$, in expression (4.11), as follows:

$$K_i = \min\left\{ W_i^{max}, W - \sum_{k=1}^{i-1} n_k \right\}, \quad i = 1, \cdots, P \tag{4.12}$$

We observe that the probability that a class-i burst would be dropped at an arbitrary time is equal to one minus the probability that class i can be allocated one wavelength at that time. Let us use $C^{-1}(W, \underline{W}^{max})$ to denote the inverse of the normalizing constant for a link with W wavelengths and vector of wavelength upper bounds $\underline{W}^{max} = (W_1^{max}, \cdots, W_P^{max})$. Also, let $\underline{1}_i$ denote a P-element vector with all elements equal to zero, except the element at position i which is equal to one. Then, the probability that a class-i burst will be dropped at an arbitrary time can be represented as:

$$B_i = 1 - \frac{C^{-1}(W-1, \underline{W}^{max} - \underline{1}_i)}{C^{-1}(W, \underline{W}^{max})} \tag{4.13}$$

Due to the Poisson arrival assumption, expression (4.13) also represents the probability that an arriving class-i burst will be dropped.

4.2.3 Wavelength Sharing with Minimum Provisioning (WS-Min)

The wavelength sharing with minimum provisioning (WS-Min) permanently allocates a number W_i^{min} of wavelengths to class i, and allows the remaining wavelengths to be shared by all classes. In other words, the wavelength lower and upper bounds are defined as:

$$0 < W_i^{min} < W_i^{max} = W, \quad i = 1, \cdots, P \tag{4.14}$$

with the additional constraint that the sum of wavelength lower bounds be less than the total number of wavelengths W, in order to allow for sharing among classes:

$$\sum_{i=1}^{P} W_i^{min} < W \tag{4.15}$$

The operation of an OBS link with the WS-Min policy is specified as follows:

> *when a class-i burst arrives to find the link at state $P(\underline{n}) = (n_1, \cdots, n_P)$, it is transmitted on any free wavelength if the number n_i of wavelengths busy with*

class-i bursts is less than the maximum number of wavelengths that class i may use at that time:

$$n_i < \max\left\{W_i^{min}, \sum_{k \neq i} \max\left\{n_k, W_k^{min}\right\}\right\} \tag{4.16}$$

Otherwise, the burst is dropped.

The evolution of the link under the WS-Min policy is described by a Markovian process whose feasible state space is defined by the following expression:

$$S = \left\{\underline{n} \mid \sum_{i=1}^{P} \max\left\{n_i,\, W_i^{min}\right\} \leq W, i = 1, \cdots, P\right\} \tag{4.17}$$

The steady state probabilities $P(\underline{n})$ have a product form similar to (4.9), and the normalizing constant C can be computed via the P-fold nested sum in (4.11). In this case, the upper limits K_i in the P-fold nested sum are given by the following expression

$$K_i \quad = \quad W - \sum_{k=1}^{i-1} \max\left\{n_k, W_k^{min}\right\} - \sum_{k=i+1}^{P} W_k^{min} \tag{4.18}$$

rather than expression (4.12). Similar to the previous subsection, let $C^{-1}(W, \underline{W}^{min})$ denote the inverse of the normalizing constant for a link with W wavelengths and vector of wavelength lower bounds $\underline{W}^{min} = (W_1^{min}, \cdots, W_P^{min})$. Then, based on our discussion regarding expression (4.13), the probability that an arriving class-i burst will be dropped at the link is given by:

$$B_i \quad = \quad 1 - \frac{C^{-1}(W-1, \underline{W}^{min} - \underline{1}_i)}{C^{-1}(W, \underline{W}^{min})} \tag{4.19}$$

4.2.4 Wavelength Sharing with Minimum Provisioning and Maximum Occupancy (WS-MinMax)

The WS-Max policy prevents any single traffic class from occupying all available wavelengths of the OBS link by imposing the wavelength upper bounds W_i^{max}. However, it cannot provide guarantees to a given class since it is possible for a few aggressive classes to consume all of the link's transmission resources. The WS-Min policy, on the other hand, can guarantee a minimum level of performance to each traffic class through the wavelength lower bounds W_i^{min}. However, it does not impose any constraints on the shared wavelengths, which may lead to unfair utilization of these resources. The policy we present in this section, namely, wavelength sharing with minimum provisioning and maximum occupancy (WS-MinMax), combines the features of the WS-Max and WS-Min policies to provide per-class QoS guarantees and high link utilization.

The WS-MinMax policy reserves a number W_i^{min} wavelengths to be used exclusively by class i, but it also restricts the number of wavelengths that can be occupied simultaneously by class-i bursts to W_i^{max}:

$$0 \;<\; W_i^{min} \;<\; W_i^{max} \;<\; W, \quad i = 1, \cdots, P \tag{4.20}$$

In addition, the following constraints are imposed on the wavelength lower and upper bounds to ensure a certain level of wavelength sharing among the traffic classes:

$$\sum_{i=1}^{P} W_i^{min} \;<\; W, \qquad \sum_{i=1}^{P} W_i^{max} \;>\; W \tag{4.21}$$

Since WS-MinMax is a generalization of both WS-Min and WS-Max, its operation can be described as follows:

> *when a class-i burst arrives to find the link at state $P(\underline{n}) = (n_1, \cdots, n_P)$, it is transmitted on any free wavelength if the number n_i of wavelengths busy with class-i bursts is less than the maximum number of wavelengths that class i may use at that time:*
>
> $$n_i < \min \left\{ W_i^{max}, W - \sum_{k \neq i} \max \left\{ n_k, W_k^{min} \right\} \right\} \tag{4.22}$$
>
> *Otherwise, the burst is dropped.*

With the WS-MinMax policy, the set of feasible states of the Markovian process describing the evolution of the OBS link becomes:

$$S = \left\{ \begin{array}{c} \underline{n} \mid 0 \leq \sum_{i=1}^{P} \max \left\{ W_i^{min}, n_i \right\} \leq W, \\ 0 \leq n_i \leq W_i^{max}, i = 1, \cdots, P \end{array} \right\} \tag{4.23}$$

Let $C^{-1}(W, \underline{W}^{min}, \underline{W}^{max})$ denote the inverse of the normalizing constant for an OBS link with W wavelengths and vectors of lower and upper wavelength bounds $\underline{W}^{min}$ and $\underline{W}^{max}$, respectively. The normalizing constant is computed by the P-fold nested sum (4.11), where the upper limits K_i in the sum are given by:

$$K_i = \min \left\{ W_i^{max}, W - \sum_{k=1}^{i-1} \max \left\{ n_k, W_k^{min} \right\} - \sum_{k=i+1}^{P} W_k^{min} \right\} \tag{4.24}$$

Then, as before, the probability that an arriving class-i burst will be dropped at the OBS link is:

$$B_i = 1 - \frac{C^{-1}(W-1, \underline{W}^{min} - \underline{1}_i, \underline{W}^{max} - \underline{1}_i)}{C^{-1}(W, \underline{W}^{min}, \underline{W}^{max})} \tag{4.25}$$

4.3 Policy Optimization

In the previous section, we presented a suite of wavelength sharing policies for a single OBS link with $P, P > 1$, traffic classes. We also showed how to compute the burst drop probability for each class $i, i = 1, \cdots, P$, as a function of the offered loads ρ_i and the wavelength lower and/or upper bounds W_i^{min} and W_i^{max}, respectively. In this section, we present a method for selecting the wavelength lower and upper bounds for the guaranteed traffic classes so as to keep the burst drop probabilities below a desired threshold. In other words, our goal is to control the level of resource sharing at the link level in a near-optimal manner in order to achieve absolute QoS differentiation among the traffic classes.

As we discussed earlier, we consider an OBS link with W wavelengths and P classes of traffic. Each traffic class $i, i = 1, \cdots, P$, is characterized by a worst-case *link* (or *one-hop*) loss guarantee B_i^{ℓ}, which corresponds to the fraction of bursts from class i that are dropped by the link in the long run. We defer to the next section the issue of translating the end-to-end loss guarantees B_i^{e2e} to appropriate link loss guarantees B_i^{ℓ}. Without loss of generality, we again assume that class i has stricter QoS requirements than class $j > i$:

$$B_i^{\ell} < B_j^{\ell}, \quad 1 \le i < j \le P \tag{4.26}$$

Traffic class P, the best-effort class, has no associated worst-case loss guarantee, and we let $B_P^{\ell} = 1.0$.

Under the WP policy, the OBS link reserves $W_i(= W_i^{min} = W_i^{max})$ wavelengths for the exclusive use of class-i bursts. Let $Erl^{-1}(\rho, B)$ denote the inverse Erlang-B formula, which returns the number of wavelengths required for the burst drop probability not to exceed B, when the offered load is equal to ρ. As it was pointed out in [113], each guaranteed class i must be allocated W_i wavelengths such that:

$$W_i = Erl^{-1}(\rho_i, B_i^{\ell}), \quad i = 1, \cdots, P-1 \tag{4.27}$$

As long as the total number of reserved wavelengths, $W_{res} = \sum_{i=1}^{P-1} W_i$, is less than the total number W of wavelengths available at the link, the best-effort class, class P, will use the remaining unreserved wavelengths. If, on the other hand, $W_{res} \geq W$, then it is not feasible to carry the offered traffic mix with the given link capacity using the WP policy. In this case, it may still be possible to meet the QoS requirements of the guaranteed classes and also carry the best-effort class without additional capacity, by exploiting the statistical multiplexing gains achievable by the wavelength sharing policies, WS-Max, WS-Min, or WS-MinMax. For the remainder of this section, we will focus only on the WS-MinMax policy, as it is a generalization of both the WS-Min and WS-Max policies. The techniques we develop can be easily adapted to either WS-Min (by fixing the wavelength upper bound of each class to W) or WS-Max (by fixing the wavelength lower bound of each class to 0).

Let us assume that the OBS link operates under the WS-MinMax policy. Let ρ_i and B_i^ℓ be the offered load and link loss guarantee, respectively, of traffic class $i, i = 1, \cdots, P$ (with $B_P^\ell = 1.0$). Our objective is to determine the optimal pair of wavelength bounds (W_i^{min}, W_i^{max}) for each class so as to minimize the burst loss probability B_P of the best-effort traffic while keeping the burst loss probability B_i of each guaranteed class $i, i = 1, \cdots, P-1$ below B_i^ℓ. More formally, this optimization problem can be stated as:

minimize: B_p

subject to:

$$B_i \leq B_i^\ell, \qquad i = 1, \cdots, P-1 \tag{4.28}$$

$$0 \leq W_i^{min} \leq W_i^{max} \leq W, \qquad i = 1, \cdots, P \tag{4.29}$$

$$W_i^{min}, W_i^{max} : \text{integer}, \qquad i = 1, \cdots, P \tag{4.30}$$

and the burst drop probabilities $B_i, i = 1, \cdots, P$, are obtained from expression (4.25).

Clearly, the above is an integer optimization problem with a nonlinear objective function and nonlinear constraints (4.28). Furthermore, important mathematical properties such as monotonicity and convexity have not been established for this type of objective function [18]. Since existing optimization tools (e.g., CPLEX) are not appropriate for this problem and an exhaustive search of the entire space of candidate solutions is computationally prohibitive, in the following subsection we develop a greedy local search heuristic to obtain a near-optimal solution to this optimization problem.

4.3.1 The Local Search Heuristic

The main idea of our greedy heuristic is to attempt to decrease the value of the objective function (i.e., the burst drop probability of the best-effort class P), by slightly increasing at each iteration the burst drop probability of one of the guaranteed classes, say, class $i, i = 1, \cdots, P-1$. However, the algorithm ensures that at the end of the iteration, the burst loss guarantee of class i will not be violated. The algorithm manipulates the values of the burst drop probabilities by adjusting the wavelength lower and upper bounds of classes i and P at each iteration. For the selected guaranteed class i, in particular, the algorithm attempts to increase its burst drop probability by searching in directions which (1) reduce class i's maximum usage of wavelengths, (2) reduce class i's minimum allocation of wavelengths, or both.

More specifically, our heuristic works as follows. Let $(W_i^{min}(k), W_i^{max}(k))$ denote the pair of wavelength lower and upper bounds for class $i, i = 1, \cdots, P$, at the end of iteration $k, k = 0, 1, 2, \cdots$. Let also $B_i(k)$ denote the burst drop probability of class i at the end of the k-th iteration, as computed by expression (4.25). At the start of the $(k+1)$-th iteration, the algorithm computes the ratio $\frac{B_i(k)}{B_i^{\ell}}$ for each guaranteed class $i, i = 1, \cdots, P-1$. This ratio is a measure of how close the long-term burst drop probability of a class is to its link loss guarantee. Let m be the class for which $\frac{B_m(k)}{B_m^{\ell}}$ is minimum among all guaranteed classes. Note that the constraint in (4.28) corresponding to class m has the largest relative slack among all such constraints. In the current (i.e., $(k+1)$-th) iteration, the algorithm will modify the wavelength lower and upper bounds of classes m and P in an attempt to lower the burst drop probability $B_P(k+1)$ of the best-effort class at the expense of class-m bursts which may experience a higher drop probability $B_m(k+1)$ (the latter, however, is not allowed to exceed B_m^{ℓ}). The algorithm does not modify the wavelength lower and upper bounds of any other class during the current iteration. Note also that the criterion by which class m is selected at the beginning of each iteration reflects the greedy nature of the heuristic.

Let us now describe how the algorithm attempts to increase the burst drop probability of guaranteed class m that was selected at the beginning of the $(k+1)$-th iteration. Let $(W_m^{min}(k), W_m^{max}(k))$ be the pair of wavelength lower and upper bounds for this class at the end of the k-th iteration. At the end of the $(k+1)$-th iteration, the algorithm will determine new bounds $(W_m^{min}(k+1), W_m^{max}(k+1))$ for this class. In order to bound

$$\mathcal{L}(k+1) = \begin{cases} \{(W_m^{min}(k), W_m^{max}(k)-1), (W_m^{min}(k)+1, W_m^{max}(k)-1)\}, & W_m^{min}(k) = 0 \\ \{(W_m^{min}(k)-1, W_m^{max}(k)-1), (W_m^{min}(k)-1, W_m^{max}(k)), & \\ (W_m^{min}(k), W_m^{max}(k)-1), (W_m^{min}(k)+1, W_m^{max}(k)-1), & \\ (W_m^{min}(k)-1, W_m^{max}(k)+1)\}, & W_m^{min}(k) > 0 \end{cases} \tag{4.31}$$

---

the computational requirements of each iteration, the heuristic limits the set of candidate values for $(W_m^{min}(k+1), W_m^{max}(k+1))$ that it considers to a small neighborhood around $(W_m^{min}(k), W_m^{max}(k))$; this is the "local search" feature of the algorithm. Specifically, the local neighborhood examined during the $(k+1)$-th iteration is defined consistent with the search directions we outlined earlier as in expression (4.31), shown at the top of the page. As a result of this definition of the local neighborhood, the wavelength lower and upper bounds of class m will not be adjusted by more than one unit (up or down) at any iteration, preventing large changes in the burst drop probabilities from one iteration to the next.

For each pair (w_m^{min}, w_m^{max}) in the local neighborhood set $\mathcal{L}(k+1)$, and using the same wavelength lower and upper bounds $(W_i^{min}(k), W_i^{max}(k))$ as at the end of the the previous iteration for all guaranteed classes $i \neq m$, we determine through expression (4.25) a pair of wavelength lower and upper bounds (w_P^{min}, w_P^{max}) for the best-effort class that minimizes its burst drop probability B_P and does not violate any of the loss guarantees. Among these, we select the pairs (w_m^{min}, w_m^{max}) and (w_P^{min}, w_P^{max}) corresponding to the minimum B_P as the values for $(W_i^{min}(k+1), W_i^{max}(k+1))$ and $(W_P^{min}(k+1), W_P^{max}(k+1))$, respectively. For all other classes we let $W_i^{min}(k+1) = W_i^{min}(k)$ and $W_i^{max}(k+1) = W_i^{max}(k)$, at the end of iteration $k+1$. We then proceed with the next iteration in a similar manner. The algorithm terminates when no improvement (reduction) in the value of the objective function B_P is possible.

To fully specify the algorithm, we need to determine appropriate initial values for the wavelength lower and upper bounds of each class. While it is possible to let $W_i^{min}(0) = 0$ and $W_i^{max}(0) = W$ for each class i, doing so has two risks: it may require a large number of iterations for the algorithm to converge, and it might cause the algorithm to get trapped in a local minimum that is far away from the global optimum. Therefore, we use the information regarding the loss guarantees $B_i^\ell, i = 1, \cdots, P-1$, to start the algorithm from a more appropriate initial solution. Specifically, let W_i denote the number of wavelengths returned by the inverse Erlang-B formula (refer to expression (4.27)) for guaranteed class i. At the beginning of the algorithm, for the guaranteed classes we let:

$$W_i^{min}(0) = W_i, W_i^{max}(0) = \min\{2W_i, W\}, i = 1, \cdots, P-1 \tag{4.32}$$

while for the best-effort class we set $W_P^{min}(0)$ and $W_P^{max}(0)$ to the pair of values that minimizes $B_P(0)$ while not violating constraints (4.28).

A step-by-step description of the local search algorithm is provided in Figure 4.1. If, at the end of the main iteration, the sum of the wavelength lower bounds for the $P-1$ guaranteed classes exceed the number W of available wavelengths, then the QoS requirements cannot be met. We also note that despite the fact that, at each iteration, the algorithm proceeds in a direction that has the potential to decrease the burst drop probability of the best-effort class by increasing the corresponding probability of one of the guaranteed classes, there is no assurance that this approach will lead to a monotonic behavior in the values of the burst drop probabilities. To see this, observe that adjusting the wavelength lower and upper bounds of one traffic class affects the degree of wavelength sharing among classes, and consequently, has an effect on the burst drop probability of all classes. Furthermore, this effect is not known in advance, and can be quantified only by applying expression (4.25) with the new bounds. Nevertheless, our experimental results indicate that our algorithm converges to a local optimum after only a few iterations.

4.4 Wavelength Sharing Policies in a Network of OBS Nodes

In this section, we consider an OBS network with P traffic classes. In order to support absolute QoS guarantees, each (unidirectional) link operates under the WS-MinMax wavelength sharing policy. Since it is typical for applications to specify their QoS requirements in terms of an end-to-end burst loss guarantee, we assume that each traffic class i is associated with an end-to-end loss rate threshold B_i^{e2e}; without loss of generality, we let:

$$B_1^{e2e} < B_2^{e2e} < \cdots < B_{P-1}^{e2e} < B_P^{e2e} = 1.0 \tag{4.33}$$

WS-MinMax Policy Optimization for an OBS Link
Input: An OBS link with W wavelengths, P traffic classes, offered load ρ_i and burst loss guarantee $B_i^\ell, i = 1, \cdots, P$ ($B_P^\ell = 1.0$)
Output: Pair of wavelength lower and upper bounds $(W_i^{min}, W_i^{max}), i = 1, \cdots, P$, such that $B_i \leq B_i^\ell, i = 1, \cdots, P-1$, and B_P is minimized

procedure **PolicyOpt**
begin
$k \leftarrow 0$ // iteration index
for $i = 1$ to $P-1$ **do** // initialization
    $W_i^{min}(k) \leftarrow Erl^{-1}(\rho_i, B_i^\ell)$; $W_i^{max}(k) \leftarrow \min\{2W_i^{min}(k), W\}$
$(W_P^{min}(k), W_P^{max}(k)) \leftarrow$ pair of values that minimizes $B_P(0)$ without violating constraints (4.28)
repeat // main iteration
    $k \leftarrow k+1$
    Let m be the class with the minimum value of $\frac{B_i(k-1)}{B_i^\ell}, i = 1, \cdots, P-1$
    $\mathcal{L}(k) \leftarrow$ the local neighborhood from expression (4.31)
    $B \leftarrow 1.0$ // temporary variable
    for each $(w_m^{min}, w_m^{max}) \in \mathcal{L}(k)$ **do** // update the wavelength bounds of classes m and P
        $(w_P^{min}, w_P^{max}) \leftarrow$ pair of values that minimizes B_P without violating constraints (4.28)
        if $B_P < B$ **then**
            $W_m^{mim}(k+1) \leftarrow w_m^{min}$; $W_m^{max}(k+1) \leftarrow w_m^{max}$; $W_P^{min}(k+1) \leftarrow w_P^{min}$;
            $W_P^{max}(k+1) \leftarrow w_P^{max}$
    for $i = 1$ to $P-1, i \neq m$ **do** // wavelength bounds of other classes remain the same
        $W_i^{min}(k+1) \leftarrow W_i^{min}(k)$; $W_i^{max}(k+1) \leftarrow W_i^{max}(k)$
until B_P cannot be decreased any further
if $\sum_{i=1}^{P-1} W_i^{min} \geq W$ **then return** error // cannot meet QoS guarantees
else return $(W_i^{min}(k), W_i^{max}(k)), i = 1, \cdots, P$
end

Figure 4.1: Local search heuristic for policy optimization

The main issue we address in this section is how to optimize the parameters of the WS-MinMax policy (i.e., the wavelength lower and upper bounds of each class) at each link, so that the network will meet the end-to-end loss requirements of the guaranteed classes while minimizing the burst loss probability of the best-effort class P.

Consider any link of the network, and recall that in order to apply the policy optimization algorithm in Figure 4.1 we need to determine the link offered load ρ_i and link loss rate guarantee B_i^ℓ for each class i. The offered load ρ_i can be determined in several different ways. For instance, if the network uses fixed routing and making the reasonable assumption that link drop probabilities are relatively small), we can approximate ρ_i by summing the amount of class-i traffic offered by source-destination pairs whose path uses this link. Alternatively, the OBS node at the head of the link may periodically measure the amount of class-i traffic passing through; assuming that traffic variations take place over longer time scales, traffic measurements will yield a fairly accurate estimate of ρ_i.

Let us now turn our attention to the problem of determining the per-link loss rate guarantees B_i^ℓ from the end-to-end guarantees $B_i^{e2e}, i = 1, \cdots, P-1$. Consider the burst traffic between a certain source-destination pair and let h denote the number of links (hops) in the path. Let us further make the common assumption that link drop probabilities are independent. In this case, we can guarantee that the end-to-end loss requirement of traffic class B_i^{e2e} for this source-destination pair will be met by letting the loss thresholds at each of the h links equal to:

$$B_i^\ell(h) \;=\; 1 - \exp\left(\frac{\ln(1 - B_i^{e2e})}{h}\right), \; i = 1, \cdots, P-1 \tag{4.34}$$

Note, however, that a link may carry class-i traffic from several source-destination pairs using paths of different lengths. Let D denote the diameter of the network. One possible way of dealing with this issue would be to subdivide class-i traffic into D subclasses, where each subclass h corresponds to class-i traffic traveling over an h-link path. While theoretically possible, the computational requirements of such an approach would be prohibitive in practice, due to the explosion in the number of traffic classes involved in evaluating expression (4.25) and the corresponding increase in the running time of the policy optimization algorithm.

A simple solution to this problem was suggested in [113], where it was proposed to set the loss guarantee at each link to the value $B_i^\ell(D)$ obtained by using the diameter D of the network in place of h in expression (4.34). This approach is simple to implement and has the additional advantage that the values of B_i^ℓ are identical for all links of the network. A limitation of this method is that by using the diameter of the network in the above expression will result in over-provisioning link resources to guaranteed classes. Consequently, the network resources may not be sufficient to meet the QoS requirements of all classes, and/or the best-effort class may suffer losses that are unnecessarily high [113]. To alleviate the over-provisioning effect, it would be possible to partition the network into clusters whose diameter does not exceed a predefined threshold, and apply the above method to paths within each cluster. Maintaining multiple clusters, on the other hand, requires the use of intelligent partitioning techniques, increases complexity, and results in different per-link loss thresholds for each class.

We now propose another approach which is relatively simple to implement and specifies the same loss rate requirement B_i^ℓ at all links of the network. Let $\bar{H}$ denote the average number of hops, over all source-destination pairs, of a path in the network, and let $B_i^\ell(\bar{H})$ be the corresponding value of expression (4.34). Note that since $\bar{H} < D$, then $B_i^\ell(\bar{H}) > B_i^\ell(D)$. The first step in our approach is to check whether letting $B_i^\ell(\bar{H})$ as the per-link loss rate guarantee B_i^ℓ for class $i, i = 1, \cdots, P$. To this end, we compute the network-wide end-to-end burst loss probability of class-i traffic as [83]:

$$\mathcal{B}_i = \frac{\sum_{l \in E} B_i^\ell \times \rho_i^{(l)}}{\sum_{s,d} \rho_i^{(s,d)}}, \qquad i = 1, \cdots, P-1 \tag{4.35}$$

where E is the set of links in the OBS network, $\rho_i^{(l)}$ is the total load of class-i traffic offered to link l, and $\rho_i^{(s,d)}$ is the class-i traffic load generated by source-destination pair (s,d). If $\mathcal{B}_i < B_i^{e2e}$ for all guaranteed classes i, we let $B_i^\ell = B_i^\ell(\bar{H})$ for all links in the network, and we stop: this value of per-link loss guarantee is sufficient to meet the end-to-end QoS requirements of all classes, as well as to ensure a low value for the end-to-end loss rate of the best-effort class P.

If, on the other hand, there is some class i for which $\mathcal{B}_i > B_i^{e2e}$, then we need to impose more stringent per-link guarantees in order to meet the end-to-end QoS requirements. We now observe that the feasible values of the per-link guarantee for class i are in the range $[B_i^\ell(D), B_i^\ell(\bar{H})]$. A natural approach for searching this range of values is to perform a binary search, where at each step with let $B_i^\ell, i = 1, \cdots, P$, be the mid-point $B_i^{mid} = (B_i^{min} + B_i^{max})/2$ of the current interval $[B_i^{min}, B_i^{max}]$, where initially we let $[B_i^{min}, B_i^{max}] = [B_i^\ell(D), B_i^\ell(\bar{H})]$. If, using expression (4.35), this value B_i^{mid} is sufficient to meet the end-to-end QoS requirements, the search continues in the interval $[B_i^{mid}, B_i^{max}]$; otherwise, it continues in the interval $[B_i^{min}, B_i^{mid}]$. This binary search algorithm repeats in this manner until the length of the search range becomes sufficiently small, i.e., until $B_i^{max} \leq B_i^{min} \times \epsilon$, where $\epsilon > 1$ is a small constant. At that point, we let the per-link loss guarantee $B_i^\ell = B_i^{min}, i = 1, \cdots, P-1$.

The details of this binary search algorithm can be found in Figure 4.2. Note that for comparisons involving vectors, if any one element of the vector violates the comparison conditions, then the vector itself is assumed to also violate them.

Per-Link Loss Guarantee Optimization for an OBS Network
Input: An OBS network with diameter D and average path length $\bar{H}$, P classes of traffic, and end-to-end loss guarantee vector $\underline{B}^{e2e} = (B_1^{e2e}, \cdots, B_{P-1}^{e2e})$
Output: Per-link loss guarantee vector $\underline{B}^{\ell} = (B_1^{\ell}, \cdots, B_{P-1}^{\ell})$ such that the end-to-end loss guarantees are met and the end-to-end burst loss probability of the best-effort class is minimized

procedure **LinkGuaranteeOpt**
begin
// initialize the search range using expression (4.34)
1. $\underline{B}^{min} \leftarrow (B_1^{\ell}(D), \cdots, B_{P-1}^{\ell}(D))$
2. $\underline{B}^{max} \leftarrow (B_1^{\ell}(\bar{H}), \cdots, B_{P-1}^{\ell}(\bar{H}))$
2. **while** $\underline{B}^{max} > \underline{B}^{min} \times \epsilon$ **do** // binary search
3. $\underline{B}^{mid} \leftarrow \left(\underline{B}^{min} + \underline{B}^{max}\right)/2$
4. $\underline{\mathcal{B}} \leftarrow (\mathcal{B}_1, \cdots, \mathcal{B}_{P-1})$ from expression (4.35) with $\underline{B}^{\ell} = \underline{B}^{min}$
5. **if** $\underline{\mathcal{B}} < \underline{B}^{e2e}$ **then** // attempt to increase the link guarantees to decrease B_P
6. $\underline{B}^{min} \leftarrow \underline{B}^{mid}$
7. **else** // must decrease the link guarantees
8. $\underline{B}^{max} \leftarrow \underline{B}^{mid}$
9. **end while**
10. **return** $\underline{B}^{min}$
end

Figure 4.2: Binary search algorithm for selecting the per-link loss guarantees

4.5 Numerical Results

4.5.1 Policy Optimization at a Single OBS Link

Let us first consider a single OBS link with $W = 32$ wavelengths and $P = 3$ classes of traffic. Classes 1 and 2 require a link loss guarantee $B_1^{\ell} = 10^{-3}$ and $B_2^{\ell} = 10^{-2}$, respectively. While there are no guarantees associated with best-effort class 3, it is desirable to keep its burst drop probability as low as possible provided that doing so does not lead to a violation of the QoS requirements of the two priority classes.

In this subsection, we compare two policies in terms of their effectiveness in meeting the above objective:

1. The WP policy, described in Chapter 4.2.1 and also considered in [113], reserves W_i wavelengths for the exclusive use of class-i bursts. For each guaranteed class $i, i = 1, 2$, the number W_i of wavelengths is determined by the inverse Erlang-B formula (4.27).

2. The WS-MinMax policy, described in Chapter 4.2.4, which associates a pair of wave-

length lower and upper bounds (W_i^{min}, W_i^{max}) with each traffic class. The values of these bounds are obtained by running the policy optimization algorithm in Figure 4.1.

Figure 4.3 plots the burst drop probability against the link load ρ, in Erlang, for the three classes of traffic under the two policies, WP and WS-MinMax; it also plots the average burst drop probability over all three classes of traffic. For this figure, we assume that class-1 (respectively, class-2) bursts represent 20% (respectively, 30%) of the traffic, and the remaining traffic is best-effort; in other words, $\rho_1 = 0.2\rho$, $\rho_2 = 0.3\rho$, and $\rho_3 = 0.5\rho$. As we can see, both policies ensure that the burst loss rate for classes 1 and 2 is kept below the loss requirement of 10^{-3} and 10^{-2}, respectively. On the other hand, the burst loss for class 3 increases with the link load ρ, as expected. But whereas class 3 burst loss under the WP policy is quite high across all load values shown in the figure, under the WS-MinMax policy, class 3 burst loss is one to two orders of magnitude lower for low to moderate traffic loads; even at high loads, the burst loss rate of best-effort traffic under the WS-MinMax policy is one-half that under the WP policy. More importantly, the WS-MinMax policy reduces the overall burst drop rate significantly, with a corresponding substantial increase in throughput (not shown here due to space constraints).

The above result can be explained by noting the two main shortcomings of the WP policy. First, the policy does not allow any statistical multiplexing: it partitions the available link capacity into three sets of wavelengths, each dedicated to carrying bursts in one of the three traffic classes. The WS-MinMax policy, on the other hand, is much more flexible in allocating the link capacity to the three traffic classes. Although it does dedicate a number of wavelengths (equal to the wavelength lower bound) to each of the two guaranteed classes, it does allow for a certain degree (as determined by the policy optimization algorithm in Figure 4.1) of wavelength sharing among the three classes. The corresponding statistical multiplexing gains contribute to a decrease in the burst loss rate of best-effort, as well as overall, traffic. Hence, the WS-MinMax policy is significantly more efficient and effective in utilizing the available network resources than WP.

A second problem is that the WP policy allocates bandwidth at the granularity of a whole wavelength; as a result, it often overprovisions the guaranteed classes. This is evident from the behavior of the burst loss curves for the guaranteed classes under the WP policy in Figure 4.3. Consider, for instance, class 1. As we can see, the burst loss initially increases with the link load, but when the load goes from 21 to 21.5 Erlang, the burst loss drops. This behavior is due to the fact that up to 21 Erlang, the WP policy allocates a certain number w wavelengths to class 1 traffic, but at 21.5 Erlang it allocates $w + 1$ wavelengths. In this case, the same number $w + 1$ wavelengths are allocated for loads greater than 21.5 Erlang, hence the burst loss for class 1 continues to increase after the drop. Similar observations can be made for the burst loss curve of class 2. The WS-MinMax policy, on the other hand, by virtue of the wavelength sharing it allows, is able to allocate the link capacity at a finer granularity than a whole wavelength. Consequently, it "allocates" just enough capacity to each of the guaranteed classes to meet their loss requirements. Observe also that the burst loss for the guaranteed classes is generally higher under the WS-MinMax policy than under WP. In essence, the WS-MinMax policy reduces the loss rate of best-effort traffic by increasing the loss rate of the guaranteed classes just enough, so as not to violate the corresponding requirement.

For Figure 4.4, we fix the class 1 and class 2 load to $\rho_1 = 4$ Erlang and $\rho_2 = 6$ Erlang, respectively. The figure plots the burst loss rate of all classes under the WP and WS-MinMax policies against the load ρ_3 of the best-effort class, as the latter varies from 10 to 16.5 Erlang. Since the load of the guaranteed classes is constant, the WP policy allocates them the same number of wavelength regardless of the load of best-effort traffic; as a result, the burst loss of the two guaranteed classes is the same under the WP policy across the range of ρ_3 values. The WS-MinMax policy, on the other hand, adjusts the wavelength lower and upper bounds of the two guaranteed classes depending on the value of ρ_3, hence the behavior of the corresponding burst loss curves is non-monotonic. As a result, the WS-MinMax policy is able to reduce significantly the overall loss rate, and that of the best-effort traffic, without violating the loss requirements of the guaranteed classes.

Finally, in Figure 4.5 we fix the load of best-effort traffic to $\rho_3 = 11$ Erlang, and we also let $\rho_2 = 1.5\rho_1$. The figure plots the burst drop probability of the three classes under the WP and WS-MinMax policies as the load $\rho_1 + \rho_2$ of guaranteed traffic increases from 9 to 13 Erlang. Although the load of best-effort traffic is constant, its burst loss increases as the amount of guaranteed traffic increases, since both policies allocate additional wavelengths to the guaranteed classes. However, we again observe the significant improvement in the performance of best-effort and overall traffic under the WS-MinMax policy.

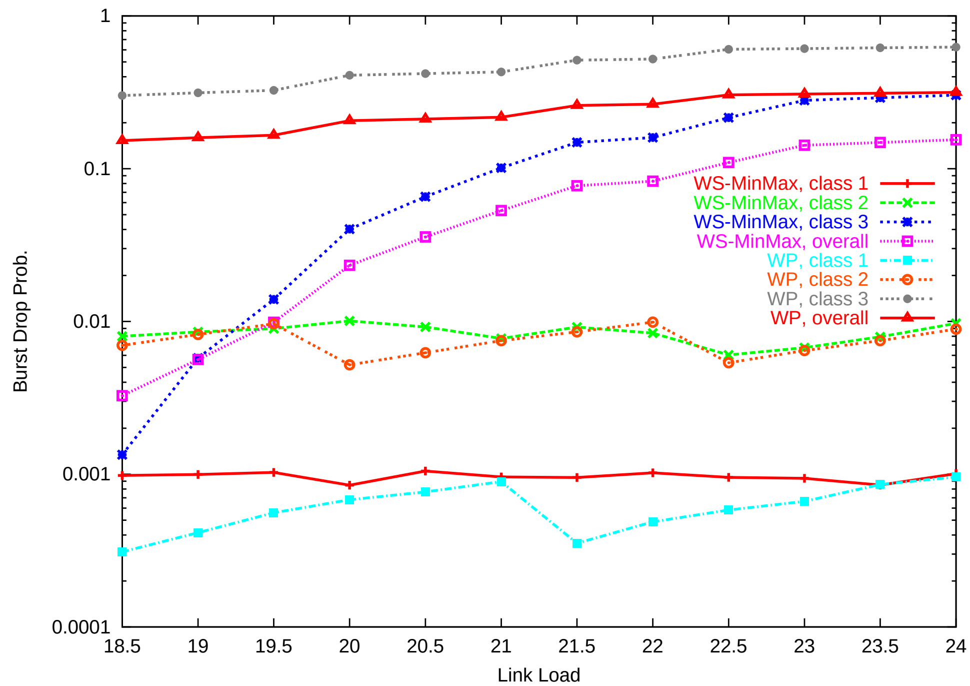

Figure 4.3: Single link with $W = 32$ wavelengths and $P = 3$ traffic classes, $\rho_1 = 0.2\rho, \rho_2 = 0.3\rho, \rho_3 = 0.5\rho$

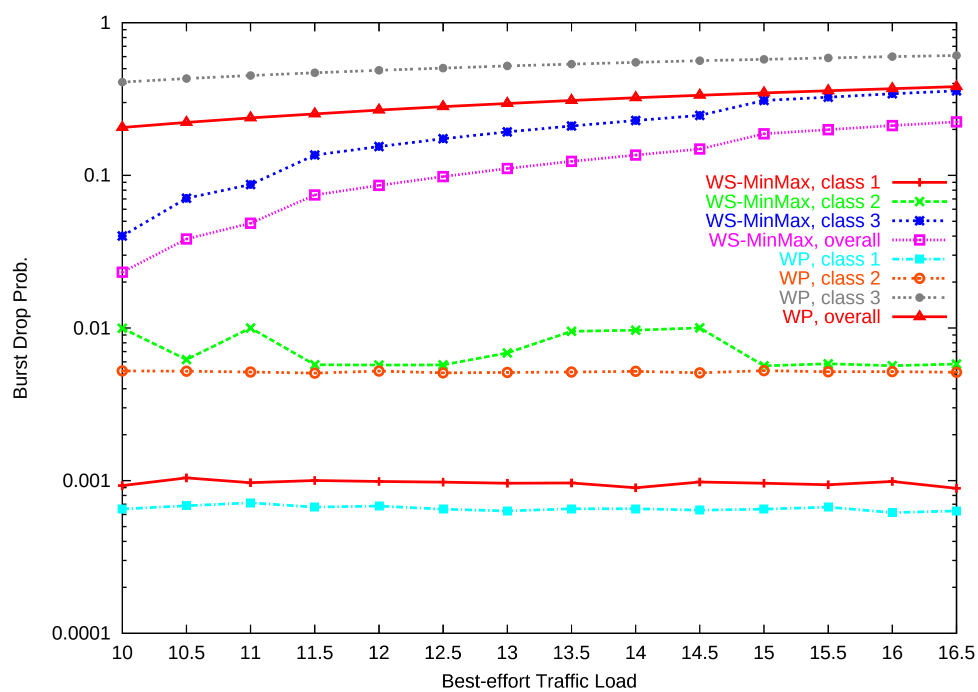

Figure 4.4: Single link with $W = 32$ wavelengths and $P = 3$ traffic classes, $\rho_1 = 4$ Erlang, $\rho_2 = 6$ Erlang

4.5.2 End-to-End QoS Guarantees in an OBS Network

We now use simulation to demonstrate the effectiveness of our wavelength sharing policies to provide end-to-end guarantees. The simulator for the experiments is discussed in 3.4. We use the method of batch means to estimate the burst drop probability, with each simulation run lasting until 6×10^5 bursts have been transmitted in the entire network. We have also obtained 95% confidence intervals for all our results; however, they are so narrow that we omit them from the figures we present in this section in order to improve readability.

In our study, we consider two 16-node networks: the 4×4 torus network shown in Figure 3.1 and the 16-node NSF network in Figure 3.2. We assume shortest path routing, and we consider two different traffic patterns:

- **Uniform pattern:** each switch generates the same traffic load, and the traffic from a given switch is uniformly distributed to other switches.
- **Distance-dependent pattern:** the amount of traffic between a pair of switches is inversely proportional to the minimum number of hops between these two switches.

We again assume that each link carries $W = 32$ wavelengths, and there are $P = 3$ classes of traffic. Classes 1 and 2 require an end-to-end loss guarantee $B_1^{e2e} = 10^{-3}$ and $B_2^{e2e} = 10^{-2}$, respectively; class 3 is the best-effort class and does not require any loss guarantees. We also note that the diameter of both the NSFNet and the torus networks is equal to 4, while the average hop distance of the two networks, used in the optimization algorithm in Figure 4.1, is $\bar{H}_{NSF} = 2.283$ and $\bar{H}_{torus} = 2.133$.

In Figure 4.6, we plot the overall burst drop probability, as well as that of the three classes of traffic, under the two policies, WP and WS-MinMax, for the NSFNet with the uniform traffic pattern. The results shown were obtained by setting the loss guarantee at each link of the network to the value obtained by using the diameter $D = 4$ of the network in place of parameter h in expression (4.34); this is the approach suggested in [113]. Figure 4.7 shows similar results for the torus network. Our observations regarding the relative behavior of the two policies, WP and WS-MinMax, from the two figures are similar to the ones we discussed in the previous section. Specifically, both policies guarantee that the burst loss of classes 1 and 2 is kept below the corresponding requirements, but the WS-MinMax policy achieves a burst loss for the overall and best-effort traffic that is significantly less than that under the WP policy. However, we also observe that using the diameter $D = 4$ to obtain the link-loss guarantees results in overprovisioning of the network for the guaranteed classes. Indeed, the network-wide burst loss of class 1 (respectively, class 2) is significantly less than the required guarantee of 10^{-3} (respectively, 10^{-2}).

In order to alleviate the overpovisioning problem, we used the optimization procedure in Figure 4.1 to determine an appropriate value for the link-loss guarantee $B_i^{\ell}, i = 1, 2$, given the corresponding end-to-end loss guarantee B_i^{e2e}. The simulation results are shown in Figures 4.8 and 4.9, for the NSFNet and torus networks, respectively. Comparing to Figures 4.6 and 4.7, we can see that using a higher value for B_i^{ℓ} results in a higher end-to-end burst loss probability for class 1 and class 2 bursts, as expected. However, the burst loss of the guaranteed classes is kept well below their requirements. Furthermore, the burst loss of best-effort traffic is reduced, as its bursts can use additional wavelength resources that were previously dedicated to the guaranteed traffic; as a result, the overall burst loss is also reduced.

Finally, in Figures 4.10 and 4.11 we present results for the torus topology with the distance-dependent traffic pattern; for the former figure, the link-loss guarantees were obtained from (4.34) with $h = D = 4$, while for the latter, they were obtained by the optimization procedure in Figure 4.1. Similar results were obtained for the NSFNet in Figures 4.12-4.13. We observe the same relative behavior for the different curves, as before; the only difference is that, due to the nature of the traffic pattern, the network can sustain the same overall burst drop probability at significantly higher traffic load compared to the uniform traffic pattern.

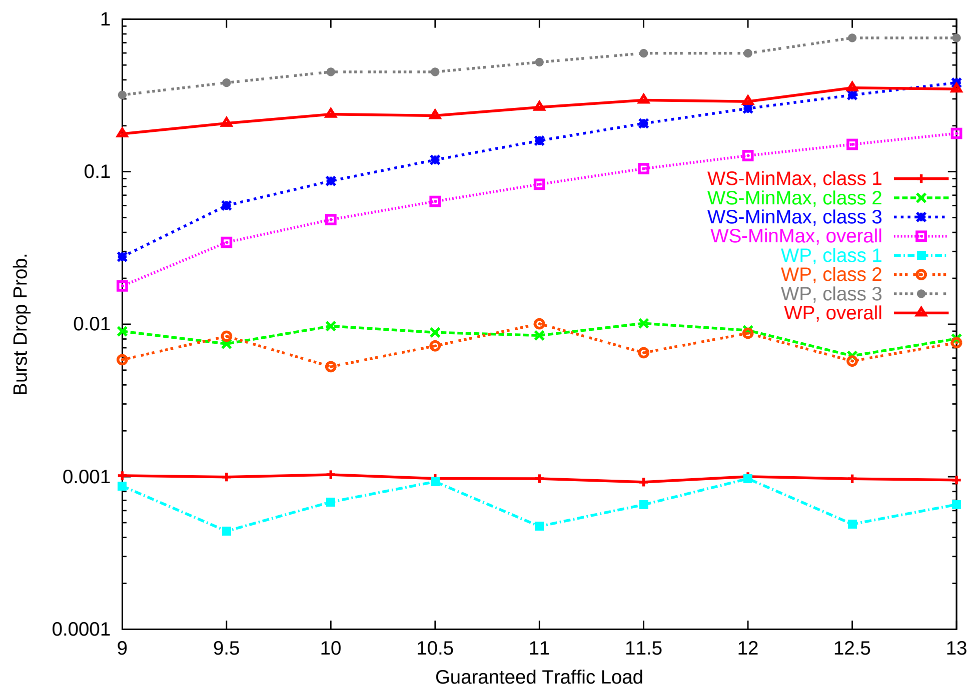

Figure 4.5: Single link with $W = 32$ wavelengths and $P = 3$ traffic classes, $\rho_3 = 11$ Erlang, $\rho_2 = 1.5\rho_1$

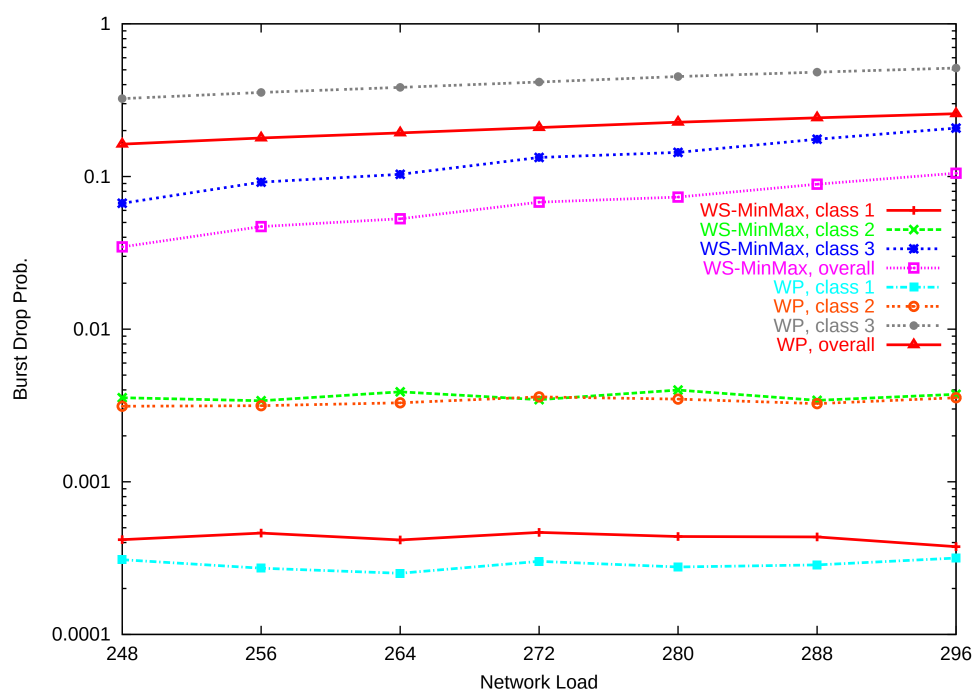

Figure 4.6: NSFNet, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained from (4.34) with $h = D$

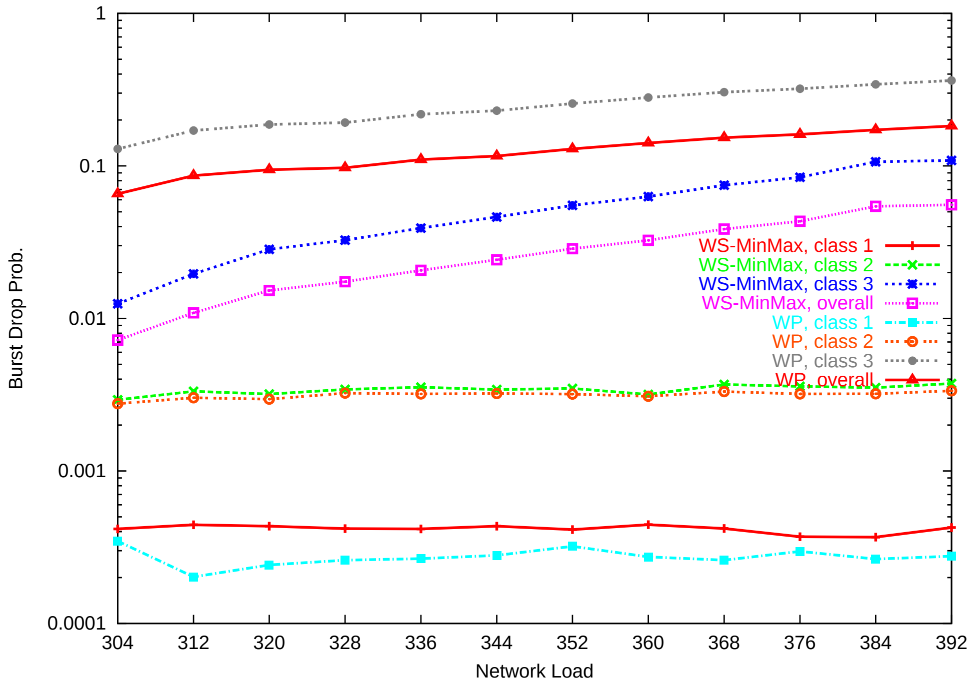

Figure 4.7: Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained from (4.34) with $h = D$

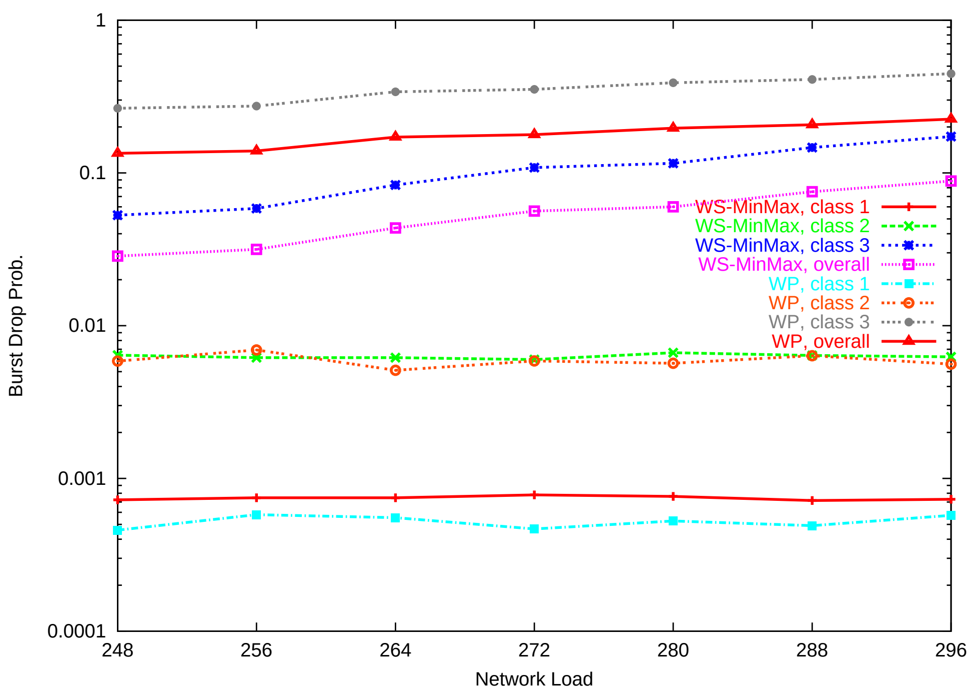

Figure 4.8: NSFNet, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained by the optimization procedure in Figure 4.1

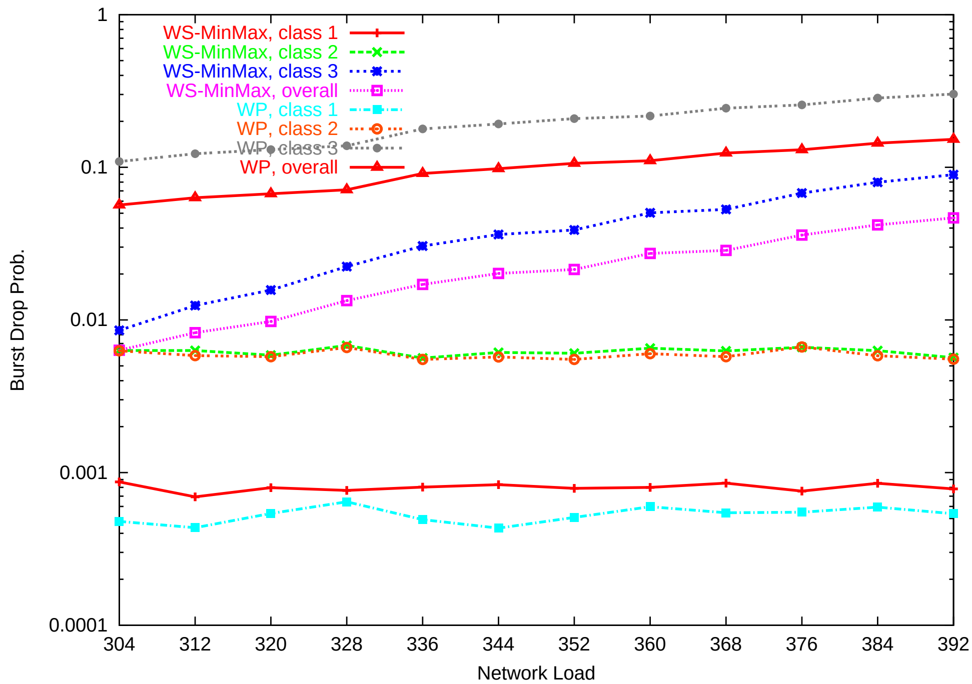

Figure 4.9: Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained by the optimization procedure in Figure 4.1

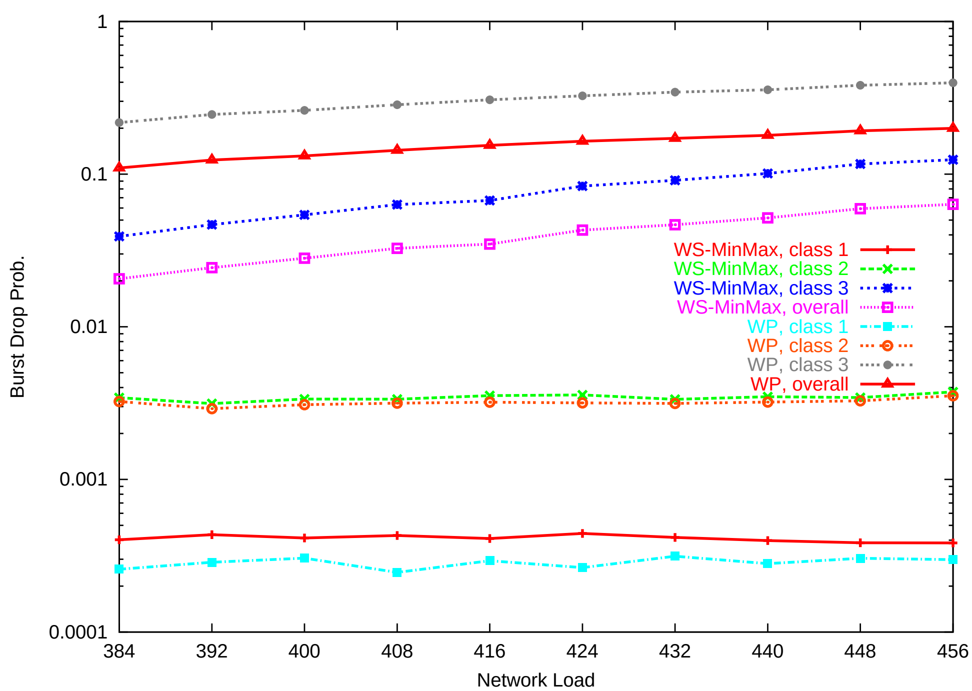

Figure 4.10: Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, distance-dependent pattern, B_i^ℓ obtained from (4.34) with $h = D$

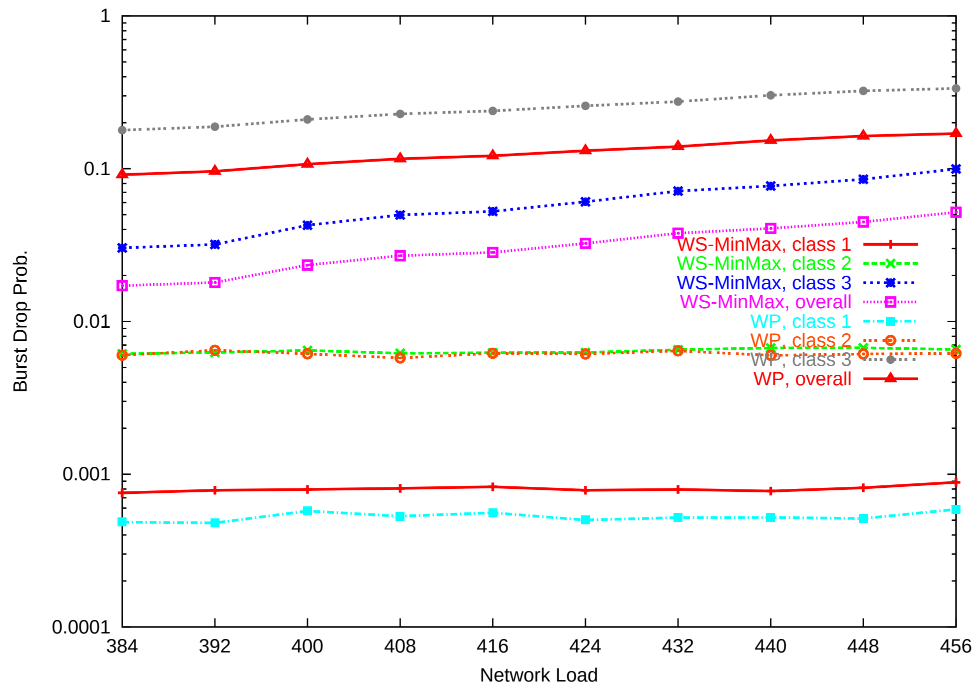

Figure 4.11: Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, distance-dependent pattern, B_i^ℓ obtained by the optimization procedure in Figure 4.1

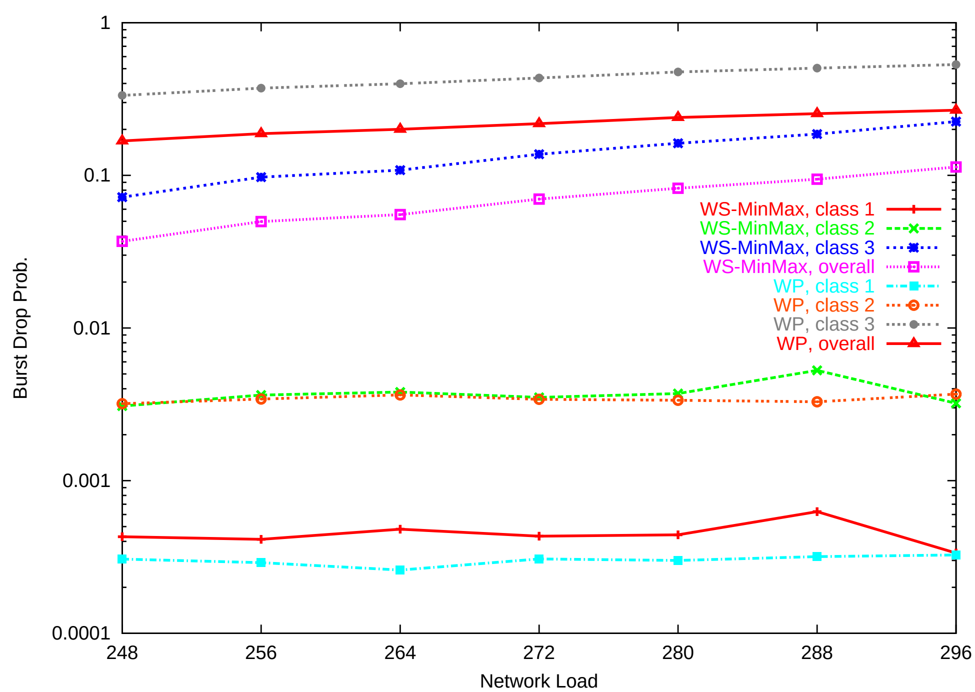

Figure 4.12: NSFNet, $W = 32$ wavelengths, $P = 3$ traffic classes, distance-dependent pattern, B_i^ℓ obtained from (4.34) with $h = D$

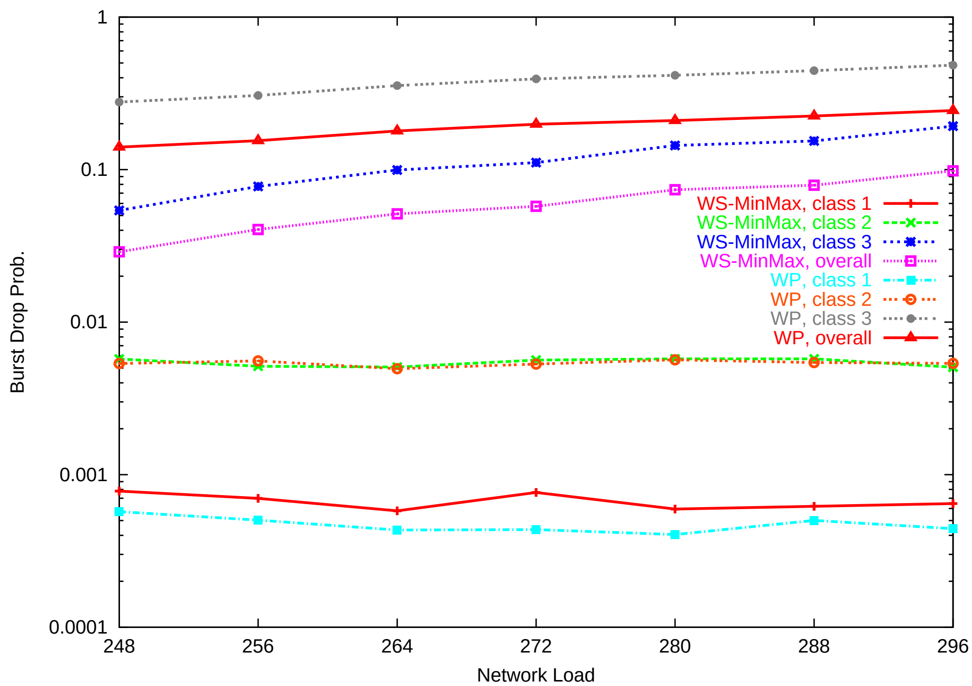

Figure 4.13: NSFNet, $W = 32$ wavelengths, $P = 3$ traffic classes, distance-dependent pattern, B_i^ℓ obtained by the optimization procedure in Figure 4.1

Chapter 5

Absolute QoS Guarantees in OBS Networks With CMDP

The purpose of this chapter is to develop a per-link wavelength provisioning scheme based on Constrained Markovian Decision Processes (CMDP) to provide service differentiation. The service differentiation is evaluated with two objectives: to maximize the throughput subject to QoS constraints; and to minimize the loss of the best effort traffic subject to the constraints on the priority traffics. We show that the CMDP approach is simple to implement and operate, and makes effective use of statistical multiplexing.

This chapter is organized as follows. In Chapter 5.1, we discuss the assumptions regarding the OBS network we consider in this study. In Chapter 5.2 we describe the model for a single OBS link. In Chapter 5.3 we introduce the model for weighted throughput maximization. In Chapter 5.4 the model for the constrained throughput maximization in a single OBS link is presented, which is applied to the whole network in Chapter 5.5. In Chapter 5.6 we present numerical results to validate the CMDP approach.

5.1 The OBS Network Under Study

Our assumptions regarding the OBS network are similar to those in Chapter 4. The detailed definition of the OBS network and the traffic classes can be found in Chapter 4.1. In addition to the worst-case *end-to-end* loss guarantee B_j^{e2e} defined for class j traffic, each class is also associated with a weight r_j, which is a measure of the importance of this class to the network. We assume that $r_j \geq r_{j+1}$ for $1 \leq j \leq P-1$, a reasonable assumption since higher priority users are likely to pay more for service.

Once assembled, a burst is assigned to one of the P classes; the mechanism for assigning bursts to traffic classes is outside the scope of our work. The class to which a burst belongs is recorded in the SETUP (control) message that precedes the burst transimission. We assume that intermediate nodes make forwarding decisions by taking into account both the availability of resources (e.g., the number of free wavelengths at an output port) and the class of a burst. Specifically, an intermediate node may drop a burst of a lower priority class even when there are wavelengths available at its outgoing link.

The objective of the network provider, and the one we consider in this work, is to:

> *ensure that the loss rate of class $i, i = 1, \cdots, P-1$, does not exceed its worst-case loss guarantee B_i^{e2e}, while at the same time maximize the weighted throughput from the network.*

Just as explained in Chapter 4.1, in order to achieve this objective, the network nodes need to employ appropriately designed mechanisms to allocate wavelength resources to bursts of each class based on its load and worst-case loss requirement. In the following, we present the wavelength reservation schemes based on MDP theory and evaluate their performance.

5.2 Model of a Single OBS Link

Let us first consider a single link of an OBS network with W wavelengths. Class j bursts arrive to the link according to a Possion process with rate λ_j. The service time of a class j burst is assumed to have an exponential distribution with mean $1/\mu_j$. The state of the link is represented by a vector $\underline{n(t)} = (n_1, \cdots, n_P)$, $\sum_{j=1}^{P} n_j \leq W$, where n_j denotes the number of class j bursts in progress on the link. The set of possible states is therefore given by

$$\mathcal{N} = \left\{ (n_1, \ \cdots, \ n_P), \ \sum_{j=1}^{P} n_j \le W \right\}. \tag{5.1}$$

The evolution of the system is described by the Markov model $\left\{\underline{n(t)}, \ t \ge 0\right\}$ and the transitions are either due to an arrival or a service completion event. We will use α_j (respectively, δ_j to denote the arrival (resp., departure) of a class$-j$ burst.

If a class j burst departs from state $\underline{n(t)}$ ($\sum_{k=1}^{P} n_k \le W,$ and $n_j > 0$), then the system moves to state $\underline{\eta(t)} = \underline{n(t)} - \underline{1}_j$ where $\underline{1}_j$ is a P-dimensional vector, with 1 at the j-th position and 0s at the remaining positions, that is $\underline{1}_j = (0, \cdots, 1, 0, \cdots)$. When a type j customer arrives to state $\underline{n(t)}$ ($\sum_{k=1}^{P} n_k \le W$), if the arrival is admitted, then the next state is $\underline{\eta(t)} = \underline{n(t)} + \underline{1}_j$; otherwise $\underline{\eta(t)} = \underline{n(t)}$. For the sake of simplicity, we will omit the index t whenever there is no ambiguity. A *control policy* determines the action to be taken at arrival events. We let $A(\underline{n}, \ \alpha_j) \in \{0, 1\}$ denote the set of actions when a class$-j$ burst arrives to find the system in state $\underline{n}$. Action $a = 0$ means that the arrival should be rejected, and $a = 1$ that the arrival should be accepted. If the system is full (all wavelengths are occupied), then the only action available is 0, thus

$$A(\underline{n}, \ \alpha_j) = 0, \quad \sum_{k=1}^{P} n_k = W, \ j = 1, \cdots, P. \tag{5.2}$$

There is no control at departure epochs, so we set

$$A(\underline{n}, \ \delta_j) = 0, \ \sum_{k=1}^{P} n_k \le W \text{ and } n_j > 0, \ j = 1, \cdots, P. \tag{5.3}$$

Since there is no control at the departure epochs, for simplicity we consider the after-departure state $\left\{\underline{\eta}\right\} = \{(n_1, \cdots, n_P)\}$ ($\sum_{j=1}^{P} n_j \le W - 1$), so that we can reduce the number of variables describing the departure events.

We consider the set of *stationary control policies* in this work. The definition of a stationary policy can be found in [38]. In essence, the controls of the stationary policy at each state are history-independent and do not change with time t. There are two commonly used types of stationary policies [38]. For a *randomized stationary policy* π defined on the state space $\mathcal{S}$, we denote by $\pi(a|s)$ as the probability that an action $a \in A(s)$ is chosen at state s; clearly, $\pi(A(s)|s) = 1, \ s \in \mathcal{S}$. If the control is a deterministic function of the state s, then the policy is called a *determinisitc stationary policy*. A randomized stationary policy π is called $k-$randomized stationary, $k = 0, 1, \cdots$, if

$$\sum_{s \in \mathcal{S}} \sum_{a \in A\{s\}} \mathbf{1}\{\pi(a|s) > 0\} \leq |\mathcal{S}| + k. \tag{5.4}$$

In other words, $k-$randomized stationary policy uses at most k actions more than a deterministic stationary policy. Apparently, a deterministic stationary policy is equivalent to a $0-$randomized stationary policy: $A(s)$ reduces to a singleton, and we use the action $\pi(s)$ at each state s to describe the policy.

5.3 Throughput Maximization in a Single OBS Link

Consider a single OBS link ℓ, and let B_j^ℓ denote the loss guarantee for class$-j$ traffic *on this link*. The quantities B_j^ℓ should not be confused with the end-to-end guarantees B_j^{e2e} in Chapter 5.1.

Our objective is to determine an optimal stationary control policy that maximizes the expected sum of the class-based rewards earned by the system. We assume that a reward r_j is collected upon accepting a burst of type j, $j = 1, \cdots, P$, while there is no reward for the departure state $\underline{\eta}$. The one-step reward for state $(\underline{n} + \underline{1}_j)$ is defined as,

$$\begin{aligned} r[(\underline{n} + \underline{1}_j),\ a = 1] &= r_j, & (5.5) \\ r[(\underline{n} + \underline{1}_j),\ a = 0] &= 0. & (5.6) \end{aligned}$$

Following the *uniformization* method in [8, Chapter 6], we can use the uniform rate $\nu = [(max_{j=1}^P \mu_j) \times W + \sum_{j=1}^P \lambda_j]$ to sample the system $\left\{\underline{n(t)}\right\}$, and get a new discrete-time Markov chain $\left\{\underline{n[i]}\right\}$, $i \geq 0$, whose sojourn times (intervals of constancy between transitions) are independently and identically distributed (i.i.d.) random variables. It has been shown that the optimal reward generated by the discrete MDP is equal to that by the original continuous time MDP [8]. A transition in the discrete system can be an arrival to state $\underline{n}$ giving a new state $\underline{\eta} = (\underline{n} + \underline{1}_j)$, a departure event leading to $\underline{\eta} = (\underline{n} - \underline{1}_j)$, or a fictitious transition from $\underline{n}$ to itself, $\underline{\eta} = \underline{n}$. Define $T(\underline{n[0]},\ \pi)$ as the expected weighted throughput when the system starts at state $\underline{n[0]}$ and follows policy π. Our goal is to find an optimal policy π^* which maximizes the expected reward over all initial states $\underline{n[0]}$,

$$\pi^* = \text{argmax}_\pi\ T(\underline{n[0]},\ \pi) = \text{argmax}_\pi \lim_{N \to \infty} \frac{1}{N+1} \left[\sum_{i=0}^{N} r(\underline{n[i]},\ a = \pi(\underline{n[i]})) \right]. \tag{5.7}$$

The transition probability $P_{\underline{n}\underline{\eta}}$ from $\underline{n}$ to state $\underline{\eta}$ is given by:

$$P_{\underline{n}\underline{\eta}} = \begin{cases} \frac{\lambda_j}{\nu}, & \text{if the arriving burst is dropped, } \underline{\eta} = \underline{n},\ a = 0,\ 1 \leq j \leq P, \\ \frac{\lambda_j}{\nu}, & \text{if the arriving burst is accepted, } \underline{\eta} = (\underline{n} + \underline{1}_j),\ a = 1,\ 1 \leq j \leq P, \\ \frac{n_j \mu_j}{\nu}, & \text{if a burst departures, } \underline{\eta} = (\underline{n} - \underline{1}_j),\ a = 0,\ 1 \leq j \leq P, \\ F, & \text{if it is a fictitious transition, } \underline{\eta} = \underline{n},\ a = 0. \end{cases} \tag{5.8}$$

where $F = \frac{\nu - \sum_{j=1}^{P} n_j \mu_j}{\nu}$ is the fictitious transition rate. We can employ the well-known Policy Iteration algorithm [8] to find an optimal policy π^* with respect to (5.7). For the special case that all traffic types have the same service times, Miller has set up a classical model for an $M/M/c/N$ system in [68] and proved that for each class the optimal policy is of threshold form, i.e., for each class j there is a critical level above which no customers of class j are admitted.

However, in the absolute QoS model we are considering, one must impose hard constraints on the worst-case service provided to the guaranteed classes. In OBS networks, these constraints are in terms of the loss experienced by each priority class. Thus, in the next section we turn our attention to the class of Constrained Markov Decision Processes, where we wish to maximize the rewards subject to the constraints on the blocking probabilities of priority classes.

5.4 Throughput Maximization in a Single OBS Link Subject to QoS Constraints

In this section we would like to determine an optimal stationary control policy that maximizes the expected sum of the class-based rewards earned by the system, subject to the constraints that the fraction of type j customers rejected is no greater than B_j^{ℓ}, $1 \leq j \leq P-1$. We assume that the service rate μ_j is equal for all classes. As discussed in Chapter 5.3 Miller established the optimality of the threshold control policy for maximizing the expected sum of class-based rewards in a $M/M/c/N$ system (similar to our OBS link), without imposing any constraints on the blocking probabilities. Feinberg and Reiman [37] extended Miller's study by adding the constraint that the blocking probability of type-1 customers not exceed a given value. They showed that for this single-constraint problem, the optimal policy has a threshold structure similiar to that in [68], but one of the thresholds may be randomized: for a particular state s, the optimal policy chooses the threshold M with probability p and the threshold $M+1$ with probability $1-p$.

5.4.1 Constrained MDP (CMDP) Formulation

Since the service rate μ_j does not depend on class j, the optimal policy can be described as a 1-Dimensional policy: we can use the total number of customers $n = \sum_{j=1}^{P} n_j$ at the i-th epoch to describe the system state [3]. Intuitively, since there is no difference in the service rates, once a burst is admitted to service, the future system evolution is not affected by the class of this burst.

Since our system does not block departures, the state $n = 0$ (corresponding to an empty system) can be reached from any other state with probability 1. Therefore, the system satisfies the *unichain* condition, which requires for every stationary policy π, the transition matrix defined by π to form a Markov chain on the state space with one ergodic class and a (possibly empty) set of transient classes. Consequently, the optimal policy is independent of the initial distribution [38].

Let us define the one-step reward and cost for the controls taken at each state. There is no reward for the departure state $\eta = 0, \cdots, W-1$. Define $r((n,\alpha_j),a), a \in A(n,\alpha_j)$ as the reward collected by the system in the arrival state (n,α_j). Specifically, we have that $r((n,\alpha_j),0) = 0$ and $r((n,\alpha_j),1) = r_j$. We define the one-step cost function c_j for class j as $c^j((n,\ \alpha_j),\ 0) = 1$, and $c^j(\eta) = c^j((n,\alpha_j),1) = 0,\ 1 \leq j \leq P$. Thus, for each rejected class$-j$ burst, the system accumulates one unit of cost. We also define the cost function C^j as the fraction of class j bursts being rejected. Since the MDP satisfies the unichain condition, the reward and cost functions are independent of the initial state.

Define the long-run time average reward earned by the system as,

$$T(\pi) \ = \liminf_{t\to\infty} t^{-1} E^{\pi}[\sum_{i=0}^{N(t)-1} r(n[i],\ a[i])]. \tag{5.9}$$

where π is a stationary policy, E^{π} is the expectation operator for the policy π, and $N(t) = \max\{i : t_i \leq t\}$ is the number of jumps by epoch t. The fraction of rejected type j customers is

$$C^j(\pi) \ = \limsup_{t\to\infty} E^{\pi}[N_j^{-1}(t) \sum_{i=0}^{N(t)} c^j(n[i],\ a[i])]. \tag{5.10}$$

where $N_j(t)$ is the number of arrivals of type j, $j = 1, 2, \cdots, P-1$ by time t. Recall that B_j^ℓ, $j = 1, \cdots, P-1$, is the loss rate to be guaranteed at the link ℓ. Then the problem of maximizing the constrained weighted throughput can be formulated as:

$$\text{maximize} \qquad T(\pi) \tag{5.11}$$

subject to

$$C^j(\pi) \quad \leq \quad B_j^\ell, \quad 1 \leq j \leq P-1 \tag{5.12}$$

In our study, we only consider the problem (5.11)-(5.12) which is feasible. Apparently, the MDP we considered does not have any absorbing state, and thus is a non-absorbing model. From the study in [39] we know there exists an optimal $K-$randomized stationary policy for feasible problems with K constraints and non-absorbing states. Since we only consider randomized stationary policies, a continous time MDP becomes an Semi-MDP in which all sojourn times have exponential distributions. It has been discussed in [38] that Linear Programming (LP) can be used to find the optimal $K-$randomized stationary policy. In the next section, we show how to make use of Linear Programming (LP) to solve the MDP problem in (5.11)-(5.12). The advantage of using LP approach is that we can incorporate additional constraints easily [75].

5.4.2 Linear Programming Formulation

First, let us introduce the following notations: $p(n, \eta, a)$ is defined to be the transition probability from state n to η if action a is taken; $z_{n,a}$ denotes the probability that action a taken at state n per unit of time. Finally, $\tau(n,\ a)$ describes the sojourn time of state $(n,\ a)$. Our objective is to find the probability $\pi(a|n)$ that an action $a \in A\{n\}$ chosen at state $n \in \mathcal{N}$, as dictated by the optimal stationary policy.

A similar constrained optimization problem is considered in [38], in which a $(P+1)$-class system with finite state space $\mathcal{S}$ and finite action set A was defined. Rewards $r_j(s,\ a)$ are collected during sojourn times for each class $j = 0, \cdots, P$ at state $s \in \mathcal{S}$. The problem studied in [38] was to maximize the reward of class 0 customers, subject to the constraints that the rewards of class j customers are no less than a given level l_j, $j = 1, \cdots, P$. Let

$s_0 \in \mathcal{S}$ as the initial state, then this problem can be expressed as,

$$\text{maximize} \quad T_0(s_0,\ \pi) \tag{5.13}$$

subject to

$$T_j(s_0,\ \pi) \geq l_j, \quad 1 \leq j \leq P. \tag{5.14}$$

To solve (5.13)-(5.14), the following LP was considered in [38]:

$$\text{maximize} \quad \sum_{s\in\mathcal{S}} \sum_{a\in A(s)} r_0(s,a) z_{s,a} \tag{5.15}$$

subject to

$$\begin{aligned}
\sum_{a\in A(s')} z_{s',a} - \sum_{s\in\mathcal{S}} \sum_{a\in A(s)} p(s,s',a) z_{s,a} &= 0, \quad s' \in \mathcal{S}, && (5.16)\\
\sum_{s\in\mathcal{S}} \sum_{a\in A(s)} r_k(s,a) z_{s,a} &\geq l_k, \quad k = 1,\cdots,P, && (5.17)\\
\sum_{s\in\mathcal{S}} \sum_{a\in A(s)} \tau(s,a) z_{s,a} &= 1, && (5.18)\\
z_{s,a} &\geq 0, \quad s \in \mathcal{S},\ a \in A(s). && (5.19)
\end{aligned}$$

It was shown that if z is the optimal solution for LP (5.15)-(5.19), then there exists a $P-$randomized stationary policy π in the form of $\pi(a'|s) = z_{s,a'} / \sum_{a\in A\{s\}} z_{s,a}$, which is optimal for problem (5.13)-(5.14).

Returning to our problem, we define $\Lambda = \sum_{j=1}^{P} \lambda_j$, then;

$$p(n,\eta,a) = \begin{cases} \frac{\lambda_j}{n\mu+\Lambda}, \\ \text{if the arriving burst is dropped, } \eta = n,\ a = 0,\ 0 \leq n \leq W, \\ \frac{\lambda_j}{n\mu+\Lambda}, \\ \text{if the arriving burst is admitted, } \eta = n + 1_j,\ a = 1,\ 0 \leq n \leq W - 1, \\ \frac{n\mu}{n\mu+\Lambda}, \\ \text{if a burst departures, } \eta = n - 1_j,\ a = 0,\ 1 \leq n \leq W. \end{cases} \tag{5.20}$$

The first case corresponds to an arriving burst being dropped, the second to an arriving burst being admitted, and the third to a burst departure. Regarding sojourn times, we have:

$$\tau(n,\ a) = \begin{cases} (n\mu + \Lambda)^{-1}, & \text{if } a = 0,\ 0 \leq n \leq W \\ ((n+1)\mu + \Lambda)^{-1}, & \text{if } a = 1,\ 0 \leq n \leq W - 1. \end{cases} \tag{5.21}$$

Then the LP corresponding to (5.11)-(5.12) to can be formulated as,

$$\text{maximize} \sum_{n\in\mathcal{N}} \sum_{a\in A(n)} r(n,a) z_{n,a} \tag{5.22}$$

subject to

$$\begin{aligned}
\sum_{a\in A(\eta)} z_{\eta,a} - \sum_{n\in\mathcal{N}} \sum_{a\in A(n)} p(n,\eta,a) z_{n,a} &= 0, \quad \eta \in \mathcal{N}, &(5.23)\\
\sum_{n\in\mathcal{N}} \sum_{a\in A(n)} c^j(n,a) z_{n,a} &\le B_j^\ell \times \lambda_j, \quad j = 1,2,\cdots,P-1 &(5.24)\\
\sum_{n\in\mathcal{N}} \sum_{a\in A(n)} \tau(n,a) z_{n,a} &= 1, &(5.25)\\
z_{n,a} &\ge 0, \quad a \in A(n),\ n \in \mathcal{N}. &(5.26)
\end{aligned}$$

Equation (5.22) represents the objective to maximize. Equation (5.23) maintains the flow balance for each state, while expression (5.24) represents the constraint in terms of the time-average burst drop rate for each class. Note that the left side of (5.24) computes the cost for each class over the time t, while the $B_j^\ell \times \lambda_j$ at the right side represents the time-average cost for comparison. Expression (5.25) simply states that the summation of the selection probability over all states and controls is equal to 1.

From the discussion in Chapter 5.4.1, we can see there exists an optimal policy π^* in the form of (5.27), where z is optimal for the LP (5.22)-(5.26), stating the probability $\pi(a|n)$ for each action a, $a \in A\{n\}$ chosen at n. The optimal policy is $P-1$ randomized, thus there are at most $P-1$ states such that $0 < \pi(a|n) < 1$.

$$\pi(a=1|n) = \begin{cases} z_{n,\ a=1} / \sum_{a=0}^{1} z_{n,\ a}, & \text{if } \sum_{a=0}^{1} z_{n,\ a} \neq 0 \\ \mathbf{1}\{a' = a\}, a' \text{ is an arbitrary element of } A(i), & \text{otherwise.} \end{cases} \tag{5.27}$$

and,

$$\pi(a=0|n) = 1 - \pi(a=1|n). \tag{5.28}$$

The optimal policy π^* works as follows. If the system state is n and a burst of class j arrives, the burst will be admitted if $\pi[(n,\ \alpha_j),\ a=1] = 1$; it will be rejected if $\pi[(n,\ \alpha_j),\ a=0] = 1$. If $0 < \pi[(n,\ \alpha_j),\ a=1] < 1$, then we draw a random number with probability θ; the burst will be admitted if $\theta \le \pi[(n,\ \alpha_j),\ a=1]$ and rejected otherwise.

5.4.3 Structure of the Optimal Policy

In [37], the authors analyzed the structure of the optimal policy which maximizes the expected average reward subject to the constraint that the blocking probability of type 1 customers is no greater than a given level. They proved that the probabilities π dictated by the optimal policy conform to expressions (5.29)-(5.31). For our problem, we have also noticed that the optimal policy has the same properties described in (5.29)-(5.31); however, we have not been able to prove this result yet.

$$\pi[(n,\ \alpha_{\{j=1\}}),\ a=1]=1, \quad n=0,\cdots W-1, \quad (5.29)$$

$$\pi[(n,\ \alpha_j),\ a=1] \geq \pi[((n+1),\ \alpha_j),\ a=1], \quad n=0,\cdots,W-2,\ j=1,\cdots,P, \quad (5.30)$$

$$\pi[(n,\ \alpha_j),\ a=1] \geq \pi[(n,\ \alpha_{\{j+1\}}),\ a=1], \quad n=0,\cdots,W-1,\ j=1,\cdots,P-1. \quad (5.31)$$

Expression (5.29) states that the bursts of class 1 (the highest priority class) are always admitted as long as there are available resources in the system. According to expression (5.30), the optimal policy is such that the probability that a class$-j$ burst will be admitted (i.e., action $a=1$ is taken) is a non-increasing function of the system occupancy n. Finally, expression (5.31) states that the probability that an arriving burst is admitted at a given state n is a non-increasing function of the burst class (i.e., bursts of lower priority have lower probability to be admitted than bursts of higher priority at a given state).

Another observation is that for each class j, there is at most one state $M_j < W$ such that $0 < \pi[(M_j,\ \alpha_j),\ a=1] < 1$; we refer to this as the *threshold state* for class j. If a class$-j$ burst arrives to find fewer than M_j bursts in the system, the burst is always accepted, and if it arrives to find more than M_j bursts, it is always rejected. If, on the other hand, the burst arrives to find exactly M_j bursts being served, then it is accepted with probability $\pi[(M_j,\alpha_j),a=1]$, and it is rejected otherwise. Similarly, expression (5.31) implies that the threshold states are such that $M_j \geq M_{j+1}, j=1,\cdots,P-1$, i.e., higher priority bursts are accepted in a larger number of states than lower priority ones.

To illustrate the structure of the optimal randomized threshold, we consider a single OBS link with $W=32$ wavelengths and $P=3$ classes of traffic. Class 1 and 2 require a link loss guarantee $B_1^\ell=10^{-3}$ and $B_2^\ell=10^{-2}$, respectively, while there are no guarantees associated with the best-effort class. We assume that class-1 (respectively, class-2) bursts represent 20% (respectively, 30%) of the traffic, and the remaining traffic is best-effort. We let the rewards r_j for admitting a class$-j$ burst take the values: $r_1=2,\ r_2=2,\ r_3=1$.

Figure 5.1 plots the thresholds for each class when the overall link load $\rho = 32$ Erlang. As we can see from the figure, since the threshold for class 1 is $M_1 = 31$ and $\pi[(M_1,\ \alpha_1),\ a = 1] = 1.0$; therefore, as long as there is a free wavelength in the system, class 1 burst will be always admitted. The threshold for class 2 is $M_2 = 29$, and $\pi[(M_2,\ \alpha_2),\ a = 1] = 0.121$. Hence, class 2 burst will be admitted if the number of bursts being serviced is less than 29; if there are exactly 29 bursts in service at the time a class-2 burst arrives, it is admitted with probability 0.121, and it is rejected with probability 0.879. The threshold for class 3 is the lowest, $M_3 = 23$, and $\pi[(M_3,\ \alpha_3),\ a = 1] = 1$; thus class 3 bursts are admitted if $n \leq M_3$.

Figure 5.2 plots the thresholds as a function of link load. Since the threshold of class 1 is always $M_1 = 31$, we only plot the thresholds of class 2 and 3, respectively. As expected, the thresholds of both classes decrease with the increase in traffic load, in order to ensure that the loss rate for class 1 does not exceed the given threshold B_1^{ℓ}.

5.4.4 Minimization of the Loss of the Best-Effort Traffic Subject to QoS Constraints

In some QoS applications, it is preferred to minimize the loss of the best-effort traffic while satisfying the requirements of the priority traffic. The primary objective of this section is to develop a mechanism that is able to provide a guaranteed loss probability for the priority traffic while also reducing the loss rate of the non-guaranteed traffic.

This problem can also be formulated as a CMDP. Referring to the formulation in Chapter 5.4, if we set $r_j = 0,\ 1 \leq j \leq P - 1$ and $r_P = 1$, then the original objective of maximizing weighted rewards is changed to maximizing the number of type P bursts accepted per unit time, which is equivalent to minimizing the number of type P bursts rejected per unit time. Since the transformation is rather straightforward, we will present the corresponding numerical results in the next section but omit the details of the formulation.

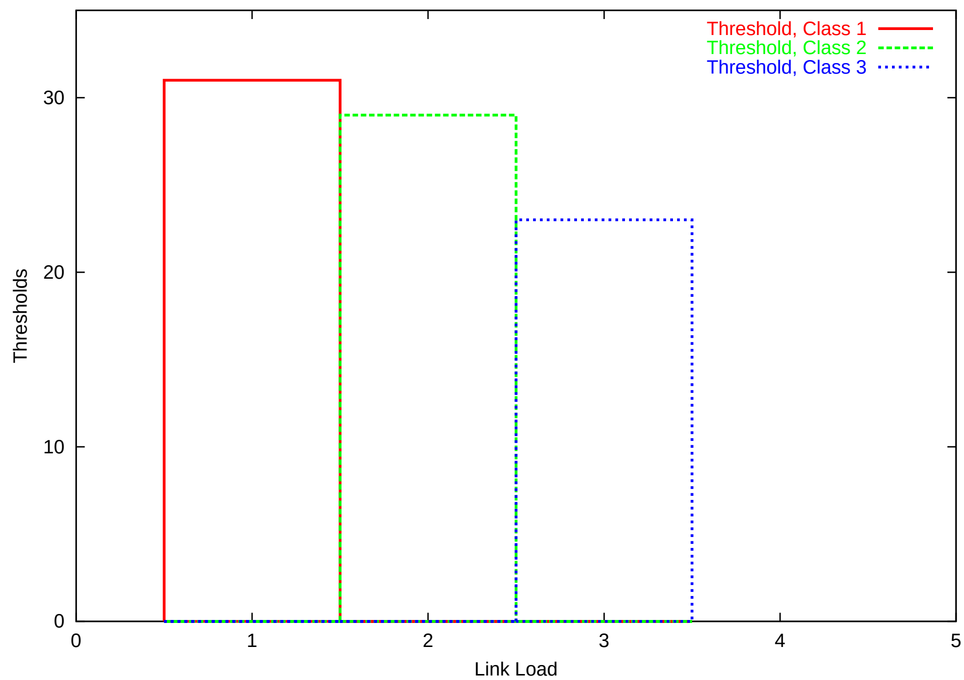

Figure 5.1: Class thresholds, Link Load = 32 Erlang

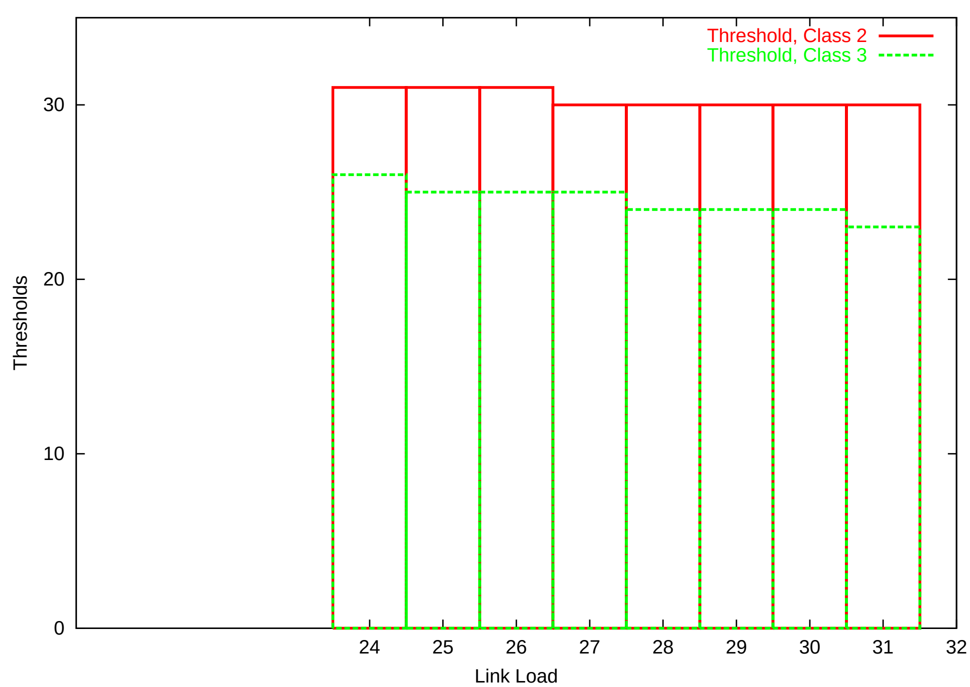

Figure 5.2: Class 1 and 2 thresholds vs. link load

5.5 Throughput Maximization Subject to QoS Constraints: The OBS Network

Typically, users (applications) are interested in the end-to-end loss, rather than loss at individual links. Assuming that the end-to-end loss guarantees B_j^{e2e} are given for all guaranteed class j, we have developed an algorithm for obtaining the link rate guarantees B_j^{ℓ} for all links ℓ in the network; due to space constraints, we omit a detailed description of the algorithm, which can be found in Chapter 4.4. The algorithm takes into account the routing paths to determine the link state guarantees such that the end-to-end guarantees are satisfied regardless of the specific path taken by the bursts. Once the value of B_j^{ℓ} are obtained for all links ℓ, we use the method we described in the previous section to obtain the optimal randomized policy for each link in the network. This approach for tackling the problem for the network as a whole, while sub-optimal, is necessiated by the fact that the state space of the Markov process describing the whole network increases exponentially with the number of links, making the problem of determining an optimal policy intractable.

5.6 Numerical Results

In this section, we first present numerical results for a single OBS link; then we apply the optimal policy to the NSF and Torus networks.

5.6.1 A Single OBS Link

Consider a link with $W = 32$ wavelengths, $P = 3$ classes of traffic, and the same loss rate guarantees, traffic mix, and reward values as in the example we used in Chapter 5.4.3 for the results in Figures 5.1 and 5.2.

Constrained Maximization of the Weighted Throughput

We compare the following two policies in terms of the overall weighted throughput that they achieve, subject to the QoS (loss rate) constraints:

1. **CMDP policy.** This is the optimal randomized threshold policy obtained through the constrained MDP (CMDP) formulation we developed in Chapter 5.4.

2. **Wavelength Partitioning (WP) policy.** The WP policy has been discussed in Chapter 4. This policy partitions the set of wavelengths into P sets (in this case, $P = 3$), and each class is given exclusive use of all wavelengths in its own set. We consider the WP policy which maximizes $\sum_{j=1}^{3} r_j \lambda_j (1 - b_j)/\mu$ subject to $b_1 \leq B_1^\ell$ and $b_2 \leq B_2^\ell$, where b_j is the blocking probability for class j under this policy.

Figure 5.3 plots the weighted throughput against the link load. The CMDP policy throughput is 15-40% higher than that of the WP policy. This result is due to statistical multiplexing: the CMDP policy makes effective use of multiplexing, but the WP policy does not allow any sharing of wavelengths among classes. Also, the CMDP throughput increases smoothly and almost linearly with the link load, whereas the WP throughput curve is non-monotonic. The latter is due to the fact that the WP policy has a granularity of one wavelength; as the load increases, it may have to shift one or more wavelengths to higher priority classes, resulting in a decrease in throughput as these wavelengths may not be utilized efficiently. The CMDP policy, on the other hand, has a much finer granularity, as it can appropriately adjust the probabilities of the threshold states for each class, giving it much more flexibility and increased efficiency in utilizing the available resources.

Constrained Minimization of the Loss of the Best-Effort Class

We compare the following two policies in terms of the loss rate of class 3 that they achieve, subject to the QoS (loss rate) constraints:

1. **CMDP policy.**

2. **The WS-MinMax policy**. The WS-MinMax policy is introduced in Chapter 4. We include the results of WS-MinMax for comparison, but as we explained in Chapter 4, the policy computed by the heuristic in Chapter 4 may not be globally optimal for the wavelength-sharing model.

We do not include the results of WP, because the WS-MinMax policy always outperforms the WP policy.

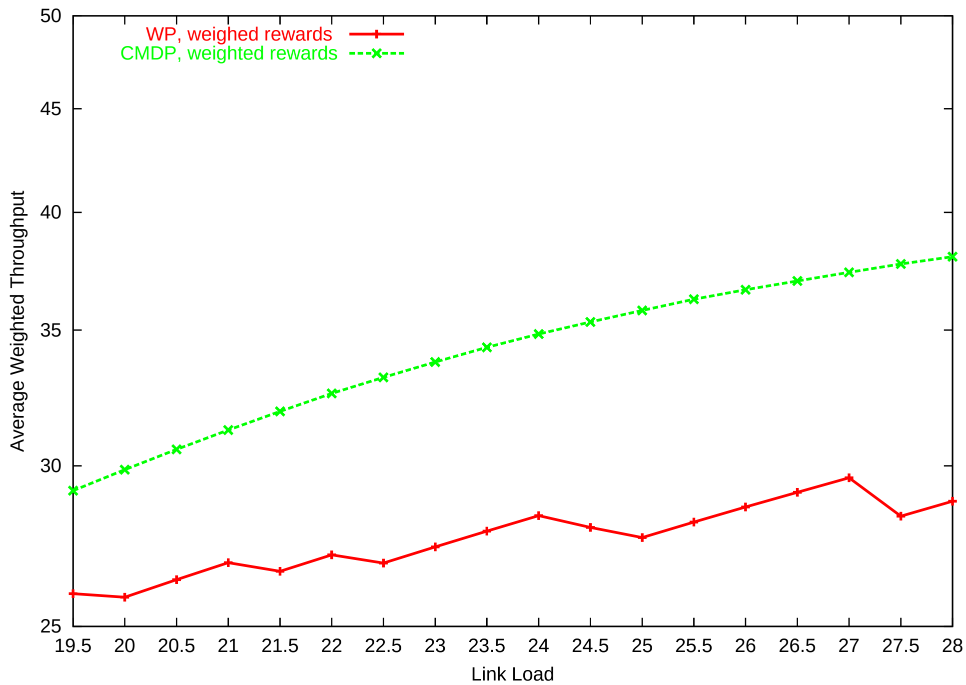

Figure 5.3: Single link with $W = 32$ wavelengths and $P = 3$ traffic classes

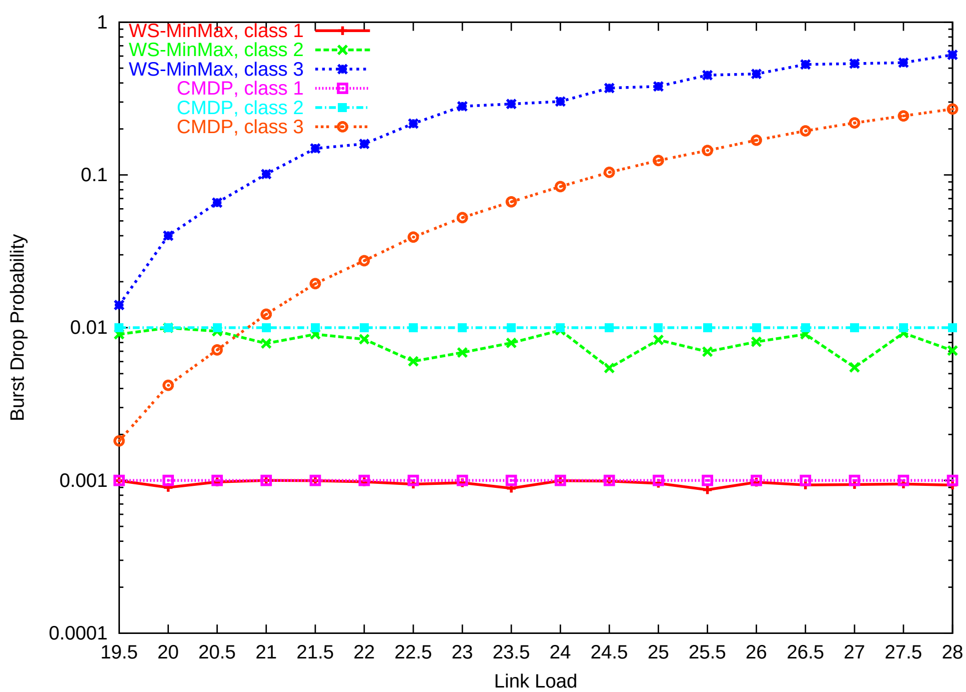

Figure 5.4: Single link with $W = 32$ wavelengths and $P = 3$ traffic classes

Figure 5.4 displays the burst drop probability of each class under the WS-MinMax and CMDP policies. As we can see, the requirements of class 1 and 2 can be guaranteed under both policies. On the other hand, the burst loss for class 3 increases with the link load ρ, as expected. But whereas class 3 burst loss under the WS-MinMax policy is the higher across all load values shown in the figure, under the CMDP policy, class 3 burst loss is 70-80% lower for low to moderate traffic loads; while at high loads, it is half that under the WS-MinMax policy.

5.6.2 The NSF and Torus Networks

We now compare the CMDP and WP policies in terms of weighted throughput on the NSF and Torus Networks. The 4×4 torus network is shown in Figure 3.1, and the NSF network in Figure 3.2. We assume shortest path routing, and we consider two different traffic patterns:

1. **Uniform pattern:** each switch generates the same traffic load, and the traffic from a given switch is uniformly distributed to other switches.

2. **Distance-dependent pattern:** the amount of traffic between a pair of switches is inversely proportional to the minimum number of hops between these two switches.

We again assume that each link carries $W = 32$ wavelengths, and there are $P = 3$ classes of traffic. Class 1 and 2 require an end-to-end loss guarantee $B_1^{e2e} = 10^{-3}$ and $B_2^{e2e} = 10^{-2}$, respectively; class 3 is the best-effort class and does not require any loss guarantees. We give the rewards r_j as $r_1 = 100,\ r_2 = 30,\ r_3 = 1$, respectively, and we included the overall burst blocking probabilities for comparison purpose.

Figures 5.5-5.6 plot the weighted throughput for the WP and CMDP policies with the uniform traffic pattern, against the network load on the NSF network. As Figure 5.5 shows, the throughput for the guaranteed classes is almost identical under the two policies. The main difference between the policies is in the throughput of the best-effort class, which is 30-100% higher under the CMDP policy, as shown in Figure 5.6. This result is due to the statistical multiplexing gains of the CMDP policy, as well as the finer granularity with which it can allocate wavelengths among the traffic classes. Also note that with the WP policy, class 3 throughput decreases as the load increases from 240-312 Erlang, and starts of increase after that. This behavior is due to the saturation of the bottleneck links: as the load increases, an increasing number of links have no wavelengths available for class-3 bursts, as resources are reserved to satisfy the QoS of guaranteed classes. Beyond 312 Erlang, the throughput increase along paths with no bottleneck links begins to dominate, and throughput starts to increase again. On the other hand, due to statistical multiplexing, the CMDP policy can provide service to the best-effort traffic even at high loads; however, class-3 throughput saturates at very high loads, as resources are needed for the guaranteed classes. Similar observations regarding the relative performance of the CMDP and WP policies can be made from Figures 5.7-5.8 which plot the weighted throughput with the non-uniform traffic pattern.

Accordingly, we plot the results on Torus network in Figures 5.9-5.10 and Figures 5.11-5.12 in both cases. And we can see the improvement of the weighted throughput for the best-effort traffic brought by the CMDP policy.

We compare the CMDP with the WS-MinMax policy regarding the reduction for the loss of best-effort class subject to the QoS constraints introduced earlier. We plot the results on NSF network with uniform traffic pattern in Figure 5.13. As we can see, CMDP can reduce the burst drop probabilities further regarding the overall traffic and the best-effort class compared with the WS-MinMax policy. The improvement in terms of loss rates is due to two reasons. First, the results of WS-MinMax policy are computed by the heuristic proposed in [104], which may not be globally optimal for the wavelength sharing model; second, the WS-MinMax policy does not belong to the optimal policies for the (constrained) reward optimization problem. However, the WS-MinMax policy is of particular interest as it can lead to product-form solutions of the steady state probabilities which give better computation complexity. From Figure 5.13 we can see that the CMDP policy can reduce the blocking probabilities of class 3 and the overall traffic by more than half compared to the WS-MinMax policy. The results with non-uniform traffic are plotted in Figure 5.14 and the improvement of the performance is demonstrated.

We also include the results on Torus network in Figure 5.15 (with uniform traffic) and Figure 5.16 (with the distance-dependent traffic pattern) with the load range indicated on the x-axis. In both cases, the CMDP policy can reduce the burst loss rates for class 3 and all traffic.

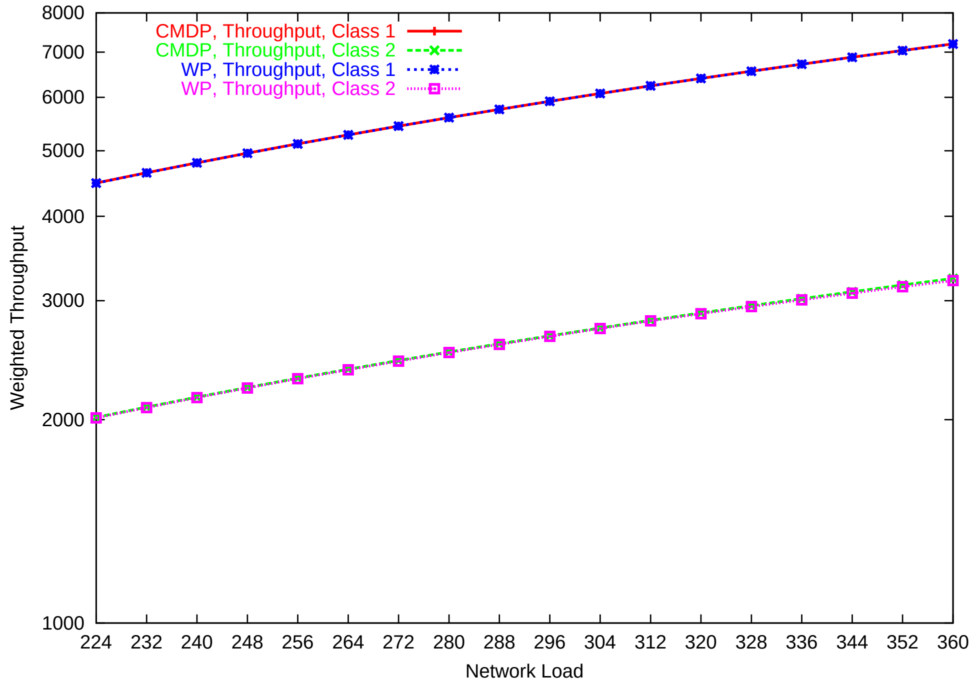

Figure 5.5: Weighted throughput on NSF network, uniform traffic pattern class-1 and class-2

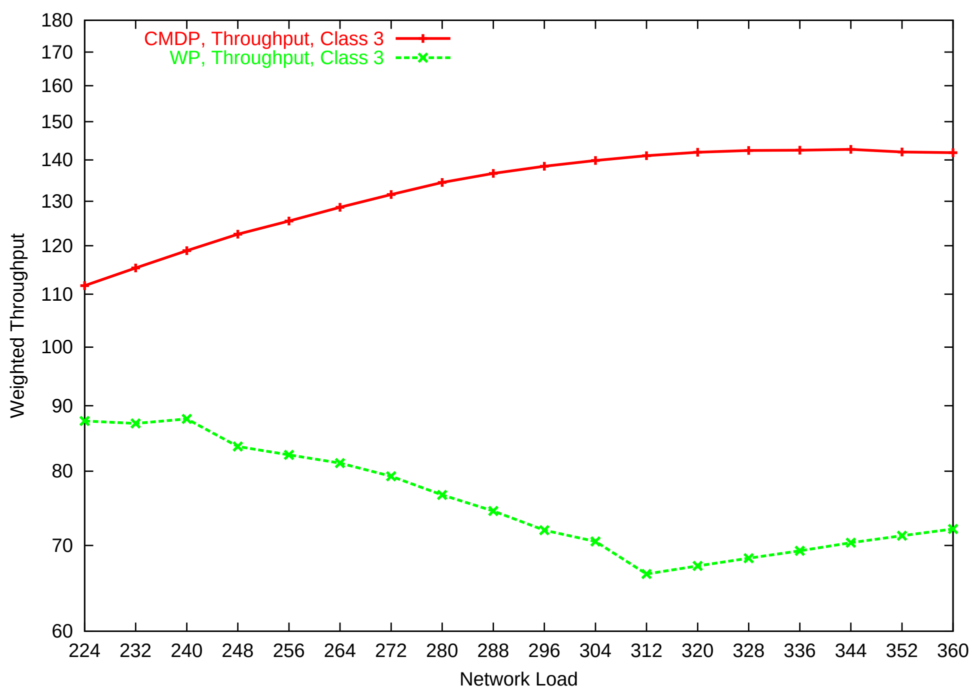

Figure 5.6: Weighted throughput on NSF network, uniform traffic pattern, best-effort traffic

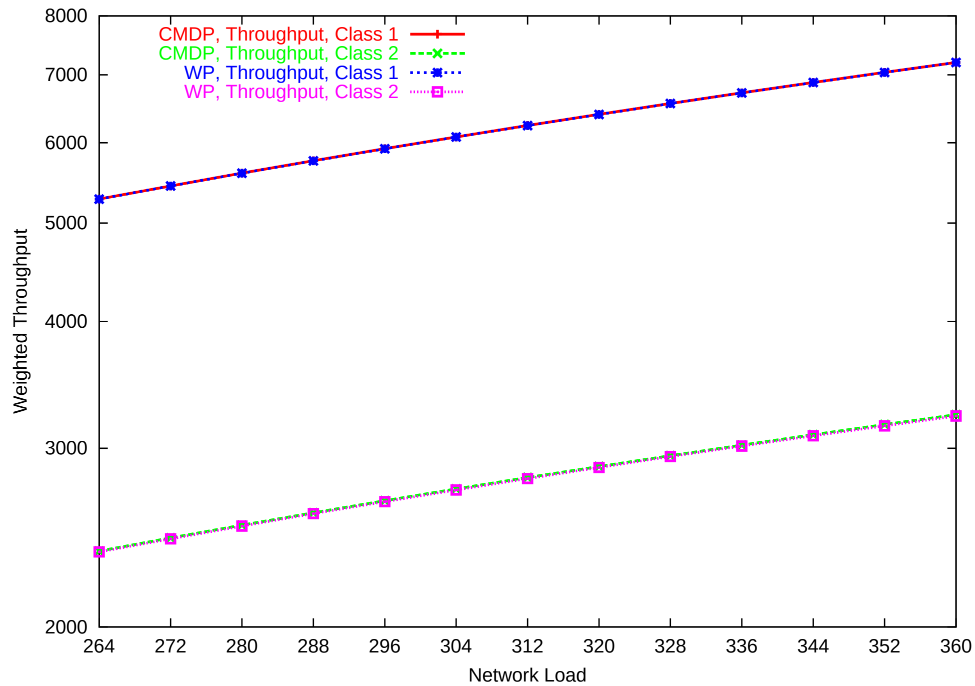

Figure 5.7: Weighted throughput on NSF network, non-uniform traffic pattern class-1 and class-2

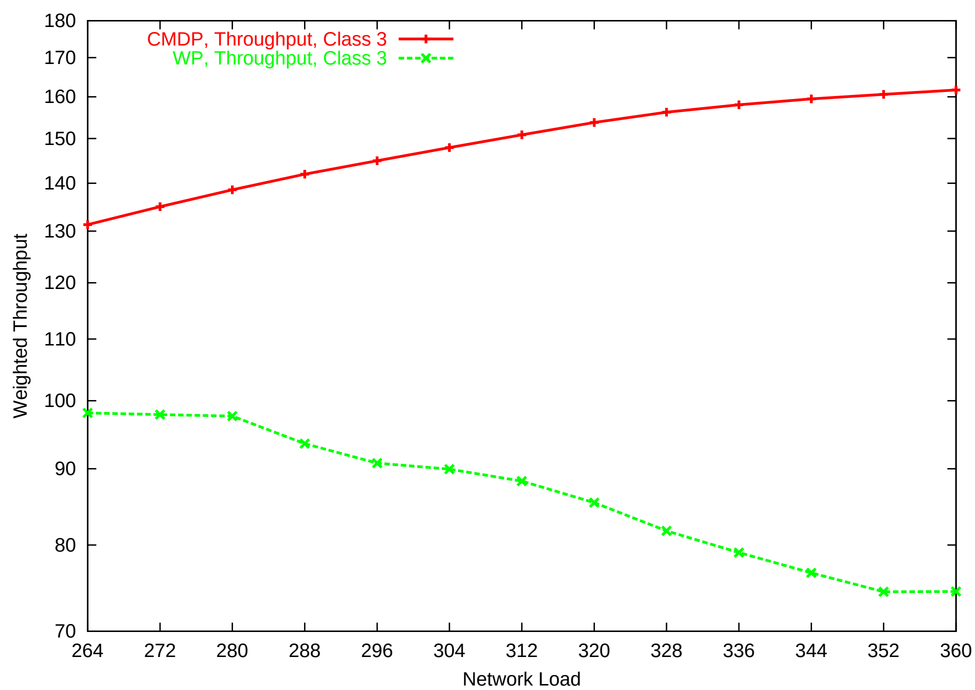

Figure 5.8: Weighted throughput on NSF network, non-uniform traffic pattern, best-effort traffic

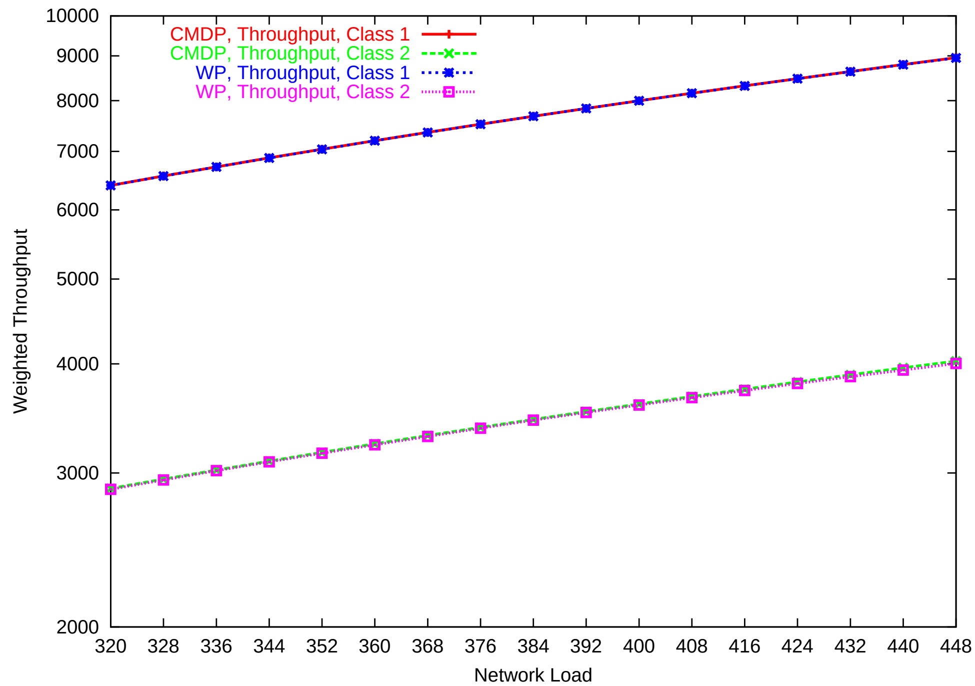

Figure 5.9: Weighted throughput on Torus network, uniform traffic pattern class-1 and class-2

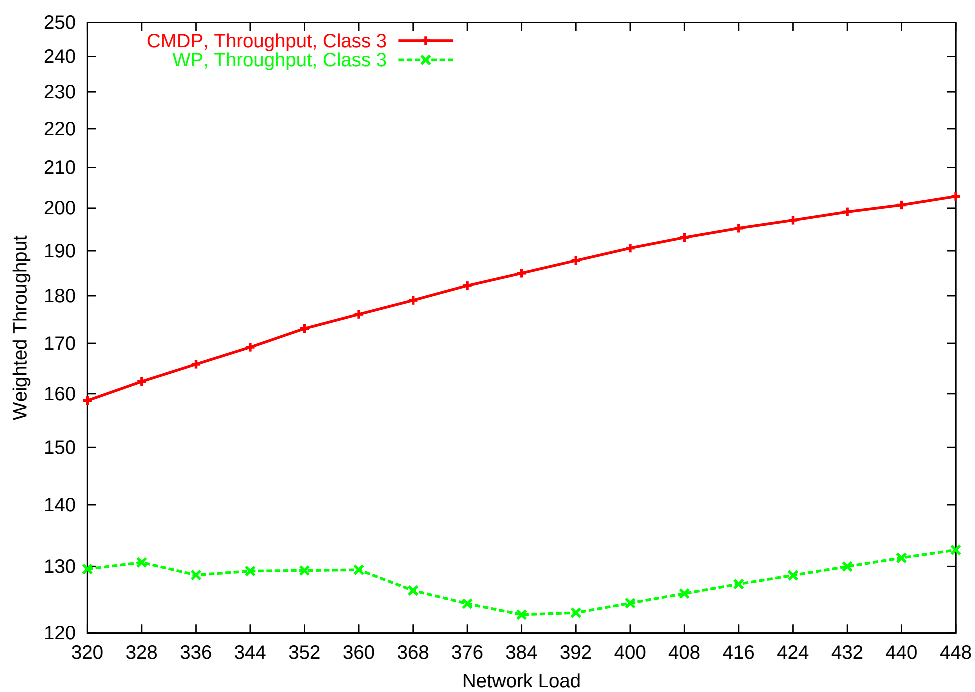

Figure 5.10: Weighted throughput on Torus network, uniform traffic pattern, best-effort traffic

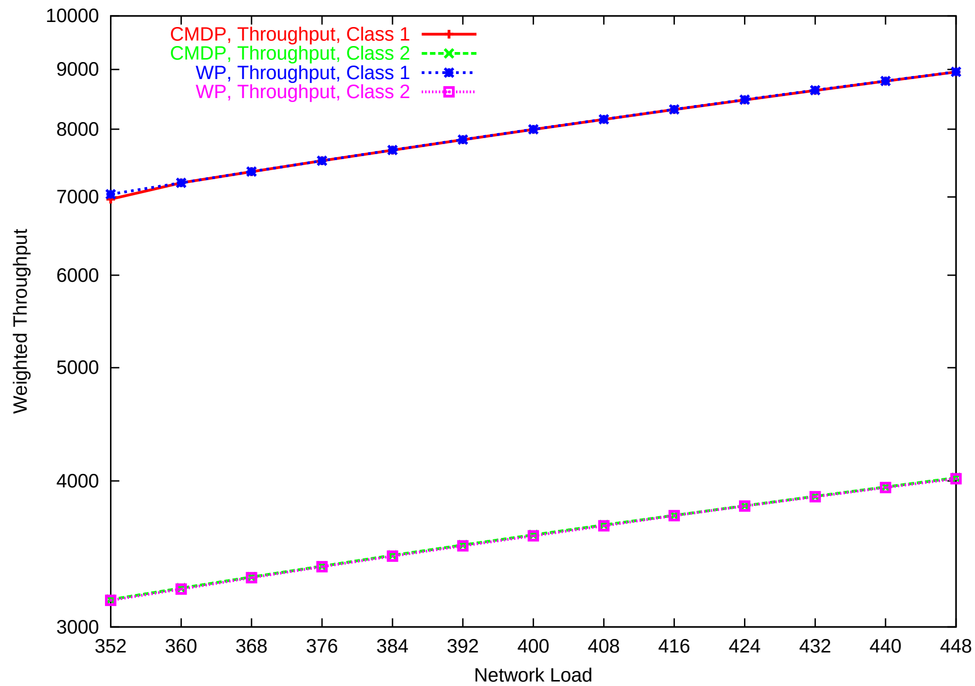

Figure 5.11: Weighted throughput on Torus network, non-uniform traffic pattern class-1 and class-2

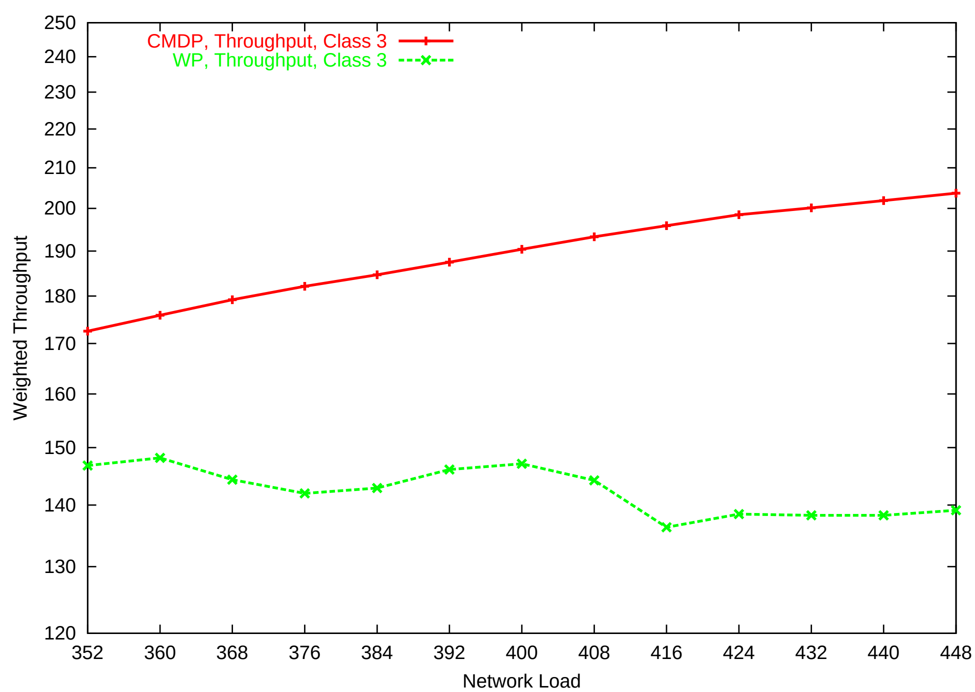

Figure 5.12: Weighted throughput on Torus network, non-uniform traffic pattern, best-effort traffic

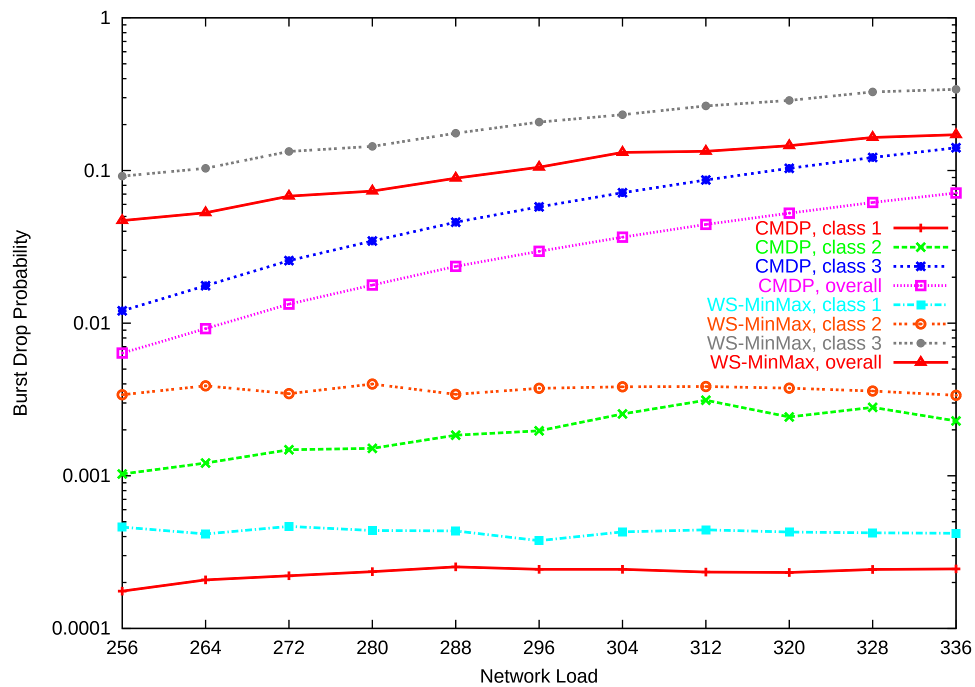

Figure 5.13: Blocking probabilities on NSF network, uniform traffic pattern

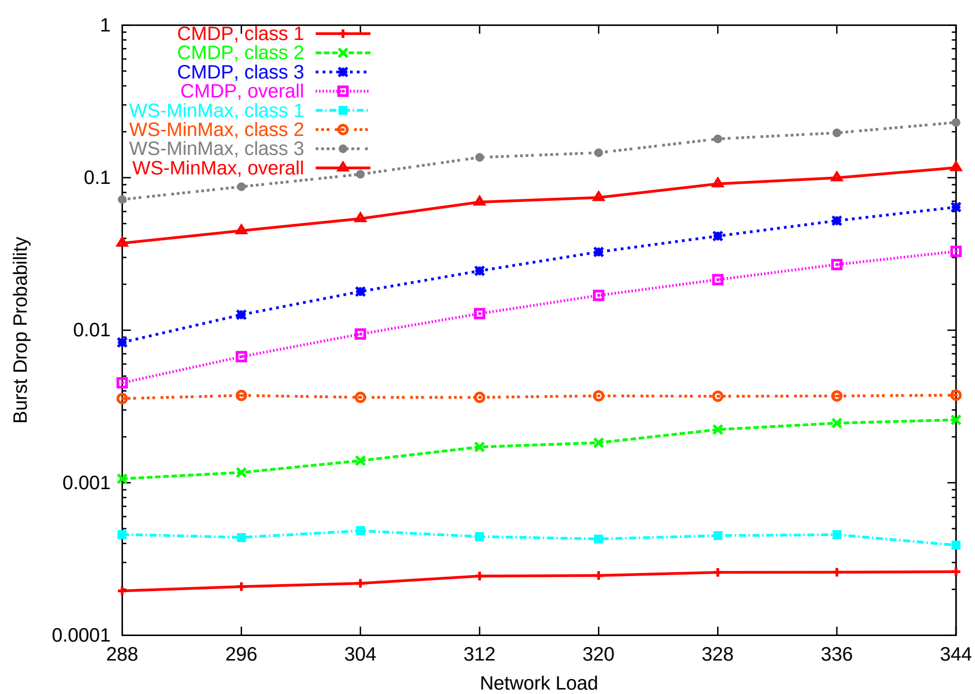

Figure 5.14: Blocking probabilities on NSF network, non-uniform traffic pattern

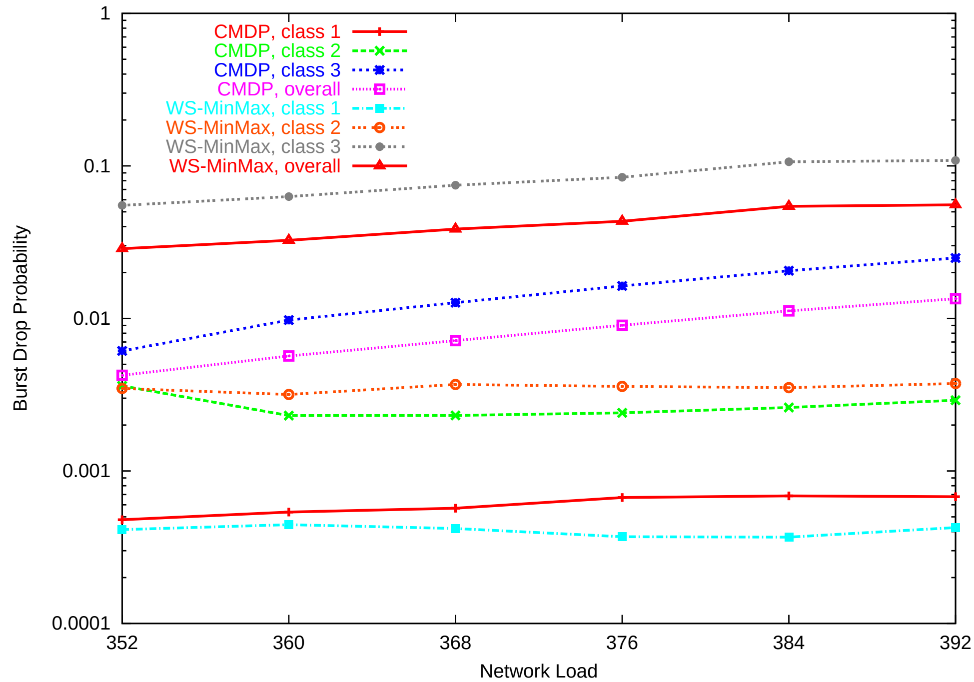

Figure 5.15: Blocking probabilities on Torus network, uniform traffic pattern

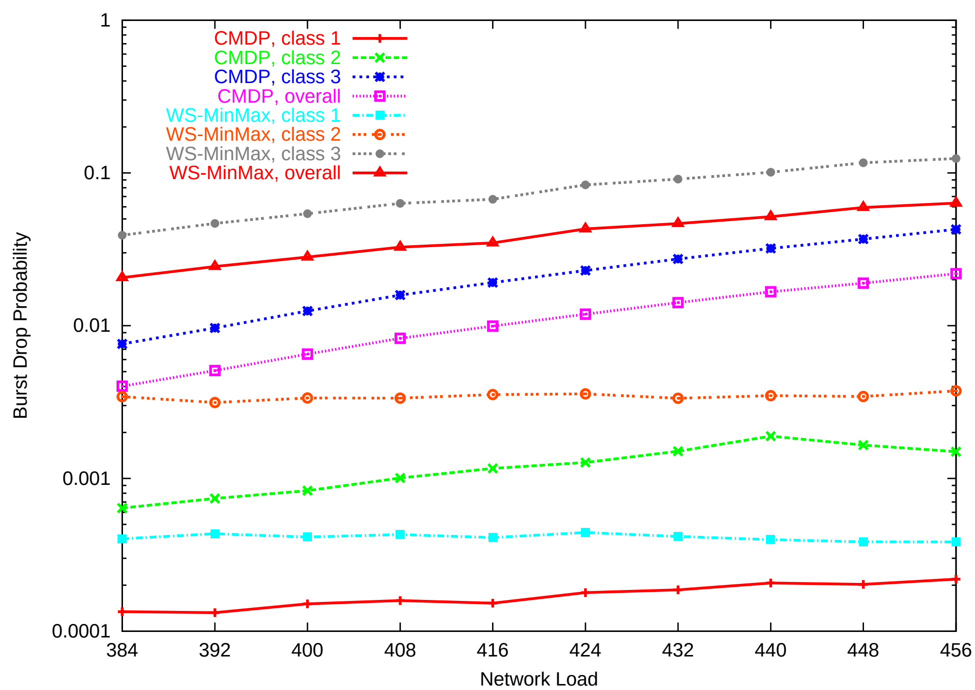

Figure 5.16: Blocking probabilities on Torus network, non-uniform traffic pattern

Chapter 6

Conclusion

This thesis studies the congestion control and quality-of-service provisioning problems in optical burst switched networks. It has mainly three parts. First it considers the problem of multipath routing in OBS networks with the objective of reducing the burst drop probability. We have developed a suite of path switching strategies, each utilizing one type of dynamic information regarding the network state to select one of a set of paths to route a given burst. We also developed a probabilistic framework for hybrid path switching strategies which make routing decisions by taking into account the decisions of multiple pure path switching strategies. We presented two instances of such hybrid strategies, one static and one dynamic. Experimental results have shown that the pure path switching strategies perform significantly better than shortest path routing, and that hybrid strategies can further improve performance in terms of burst drop probability.

Secondly this thesis presents a framework for supporting absolute QoS guarantees in OBS networks, consisting of a link wavelength sharing model, and a method to translate end-to-end loss guarantees into per-link guarantees. Our approach is effective and efficient in managing the wavelength resources, is simple to implement, and outperforms previously proposed methods.

In the third part of the thesis, we studied the wavelength reservation problem based on Constrained Markov Decision Processes in OBS network, and presented the framework and approach for calculating the constrained optimal policy. The simulation results indicate that the CMDP policy performs better than the WP method in terms of the weighted throughput; and can further reduce the best-effort traffic loss compared with the WS-MinMax policy.

6.1 Future Work

Our work can be extended in the following directions:

- The multipath selection algorithms for OBS networks can be adapted to taking into account the potential link failures. This is important for OBS networks considering the vast transport and switching capabilities of WDM, and the potential data loss as the result of a failure. In general, restoration is more efficient than protection in terms of resource utilization. Thus, efficient restoration which can reroute the affected connections over an available backup path is necessary.

- We assumed full wavelength conversion throughout this thesis. However, there are situations where full wavelength converters are not available for the whole network. Therefore, it is important to investigate sparse-partial wavelength conversion and/or limited-range wavelength converters. In this context the placement of wavelength converters is also an important problem that must be considered together with the selection of routing and wavelength assignment policies.

- The wavelength sharing policies we proposed in the second part of the thesis are burst admission control schemes with fixed thresholds. This study could be extended to adaptive policies in the sense that the optimal threshold values are automatically adjusted as operating conditions, e.g., traffic loads in the network or number of wavelengths over some links, change.

Bibliography

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

[2] N. Akar and E. Karasan. Exact calculation of blocking probabilities for bufferless optical burst switched links with partial wavelength conversion. *Proceeding of BroadNet*, Oct. 2004.

[3] E. Altman, T. Jimenez, and G. Koole. On optimal call admission control in a resource-sharing system. *IEEE Transactions on Communications*, 49(9):1659–1668, Sep. 2001.

[4] I. Baldine, M. Cassada, A. Bragg, G. Karmous-Edwards, and D. Stevenson. Just-in-time optical burst switching implementation in the ATDnet all-optical networking testbed. In *Proceedings of Globecom 2003*, volume 5, pages 2777–2781, San Francisco, USA, December 2003.

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

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

[7] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla. A performance model of deflection routing in multibuffer networks with nonuniform traffic. *IEEE/ACM Transactions on Networking*, 3(5):509–520, October 1995.

[8] Dimitri P. Bertsekas. *Dynamic Programming : deterministic and stochastic models.* Prentice-Hall, 1987.

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

[10] F. Callegati, H. C. Cankaya, Y. Xiong, and M. Vandenhoute. Design issues of optical IP routers for internet backbone applications. *IEEE Communications Magazine*, 37(12):124–128, December 1999.

[11] X. Cao, J. Li, Y. Chen, and C. Qiao. Assembling TCP/IP packets in optical burst switched networks. *Proceeding of GLOBECOM*, 3:2808–2812, Nov. 2002.

[12] H. M. Chaskar, S. Verma, and R. Ravikanth. A framework to support IP over WDM using optical burst switching. In *IEEE/ACM/SPIE Optical Network Workshop*, January 2000.

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

[14] Y. Chen and C. Qiao. Proportional differentiation: A scalable QoS approach. In *IEEE Communications Magazine*, pages 52–58, Jun. 2003.

[15] Y. Chen, C. Qiao, and X. Yu. Optical burst switching (OBS): A new area in optical networking research. *IEEE Network Magazine*, 18(3):16–23, May-Jun. 2004.

[16] G. L. Choudhury, K. K. Leung, and W. W. Att. An algorithm to compute blocking probabilities in multi-rate multi-class multi-resource loss models. *Advances in Applied Probability*, 27:1104–1143, 1995.

[17] G. L. Choudhury, K. K. Leung, and W. Whitt. Calculating normalization constants of closed queueing networks by numerically inverting their generating functions. *Journal of the ACM*, 1993.

[18] G. L. Choudhury, K. K. Leung, and W. Whitt. Efficiently providing multiple grades of service with protection against overloads in shared resources. *AT&T Technical Journal*, 74:50–63, 1995.

[19] X-W. Chu, B. Li, and Z. Zhang. A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters. In *Proceedings of IEEE INFOCOM*, pages 1795–1802, 2003.

[20] D. D. Clark and J. Wroclaski. An approach to service allocation in the internet. *IETF Draft*, Jul. 1997.

[21] D.D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *ACM Transaction on Networking*, 6(4):362–373, Aug. 1998.

[22] A. Detti and M. Listanti. Impact of segments aggregation on TCP reno flows in optical burst switching networks. Jun. 2002.

[23] T. G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18(4):97–136, April 1997.

[24] K. Dolzer and C. Gauger. On burst assembly in optical burst switching networks - a performance evaluation of Just-Enough-Time. In *Proceedings of ITC 17*, pages 149–160, 2001.

[25] K. Dolzer, C. Gauger, and J. Späth. Evaluation of reservation mechanisms for optical burst switching. *AEÜ International Journal of Electronics and Communications*, 55(1):18–26, January 2001.

[26] C. Dovrolis and P. Ramanathan. A case for relative differentiated services and the proportional differentiation model. *IEEE Network*, 13(5), Sep. 1999.

[27] C. Dovrolis and P. Ramanathan. A case for relative differentiated services and the proportional differentiation model. In *IEEE Network*, volume 13, pages 26–34, Oct. 1999.

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

[29] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *IEEE/ACM Tran. Networking*, volume 10, pages 12–26, Feb. 2002.

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

[31] M. Elhaddad, R. Melhem, T. Znati, and D. Basak. Traffic shaping and scheduling for OBS-based ip/wdm backbones. In *Proceedings of SPIE Optical Networking and Communication Conference (OptiComm) 2003*, volume 5285, pages 336–345, Oct. 2003.

[32] K. Nichols et al. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. *IETF RFC 2474*, Dec. 1998.

[33] S. Blake et al. An architecture for differentiated services. *IETF RFC 2475*, Dec. 1998.

[34] F. Farahmand and J. P. Jue. Supporting qos with look-ahead window contention resolution in optical burst switched networks. In *Proceeding of GLOBECOM*, Dec. 2003.

[35] F. Farahmand, V. M.Vokkarane, and J. Jue. Practical priority contention resolution for slotted optical burst switching networks. In *Proceedings of IEEE/SPIE First International Workshop on Optical Burst Switching (WOBS) 2003*, Oct. 2003.

[36] F. Farahmand, Q. Zhang, and J. Jue. A feedback-based contention avoidance mechanism for optical burst switching networks. In *Proceedings, 3rd International Workshop on Optical Burst Switching*, Oct. 2004.

[37] E. Feinberg and M. I. Reiman. Optimality of randomized trunk reservation. *Probability in the Engineerng and Informational Sciences*, 8:463–489, 1994.

[38] E. A. Feinberg. Constrained semi-markov decision processes with average rewards. *ZOR-Mathematical Methods of Operations Research*, 39:257–288, 1994.

[39] E. A. Feinberg. Optimal control of average reward constrained continuous-time finite markov decision processes. *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 3805–3810, Dec. 2002.

[40] W. Feng, D.D. Kandlur, D. Saha, and K.G. Shin. Adaptive packet marking for providing differentiated services in the internet. *IBM Research Report*, Jun. 1997.

[41] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions*, 7(4):458–472, Aug. 1999.

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

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

[44] R. Guerin, S. Kamat, and R. Rajan. Scalable QoS provision through buffer management. *Proceedings of ACM Sigcomm*, pages 29–40, Oct. 1998.

[45] L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, October 1990.

[46] G. Hasegawa and M. Murata. Survey on fairness issues in TCP congestion control mechanisms. *IEICE Transactions*, E00-A(1), Jan. 2001.

[47] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding PHB group. *IETF Draft*, Jan. 1999.

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

[49] J. Iness, B. Ramamurthy, B. Mukherjee, and K. Bala. Elimination of all-optical cycles in wavelength-routed optical networks. *Journal of Lightwave Technology*, 14(6):1207–1217, June 1996.

[50] V. Jacobson, K. Nichols, and K. Poduri. An expedited forwarding PHB. *IETF Draft*, Nov. 1998.

[51] R. Jain. Congestion control in computer networks: Issues and trends. In *IEEE Network Magazine*, May 1990.

[52] M. Jeong, C. Qiao, and Y. Xiong. Reliable wdm multicast in optical burst-switched networks. *Proceedings of Opticomm*, pages 153–166, 2000.

[53] M. Jeong, C. Qiao, Y. Xiong, and M. Vandenhoute. Bandwidth-efficient dynamic tree-shared multicast in optical burst-switched networks. *Proceedings of IEEE ICC*, 2:630–636, 2001.

[54] F. Kamoun and L. Kleinrock. Analysis of shared finite storage in a computer network node environment under general traffic conditions. *IEEE Transactions on Communications*, 28(7), July 1980.

[55] H. Kanakia, P. P. Mishra, and A. Reibman. An adaptive congestion control scheme for real-time packet video transport. In *SIGCOMM*, pages 20–31, 1993.

[56] J. S. Kaufman. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 29(10):1474–1481, October 1981.

[57] F. P. Kelly. *Reversibility and Stochastic Networks.* John Wiley & Sons, New York, 1979.

[58] S. Keshav. Report on the workshop on quality of service issues in high speed networks. *ACM SIGCOMM Computer Communication Review*, 22(5):74–85, Oct. 1992.

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

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

[61] J. Li and C. Qiao. Recent progress in the scheduling algorithms in optical-burst-switched networks. *Journal of Optical Networking*, 3(4), Apr. 2004.

[62] J. Li and C. Qiao. Schedule burst proactively for optical burst switching networks. *Computer Networks*, 44(5):617–629, April 2004.

[63] J. Li, C. Qiao, J. Xu, and D. Xu. Maximizing throughput for optical burst switching networks. *IEEE INFOCOM*, 2004.

[64] J. Liu and N. Ansari. Forward resource reservation for QoS provisioning in OBS systems. In *Proceedings of IEEE GLOBECOM*, 2002.

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

[66] m. Gerla and L. Kleinrock. Flow control: A comparative survey. *IEEE Transactions on Communications*, Com-28(4):553–574, May 1980.

[67] V. Mehta, N. Barakat, and E. H. Sargent. Efficient scheduling for high-throughput OBS networks with partial wavelength conversion. *Proceeding of BroadNet*, Oct. 2004.

[68] B. Miller. A queueing reward system with several customer classes. *Management Science*, 16:234–245, 1969.

[69] S. Oh and M. Kang. A burst assembly algorithm in optical burst switching networks. In *Proceedings of Optical Fiber Communication Conference and Exhibit (OFC) 2002*, pages 771–773, Mar. 2002.

[70] F. Poppe, K. Laevens, H. Michiel, and S. Molenaar. Quality-of-service differentiation and fairness in optical burst-switched networks. *Proc. SPIE Opticomm*, 4874:118–124, July 2000.

[71] C. Qiao and M. Yoo. Optical burst switching (OBS)-A new paradigm for an optical Internet. *Journal of High Speed Networks*, 8(1):69–84, January 1999.

[72] R. Ramaswami and K. N. Sivarajan. *Optical Networks*. Morgan Kaufmann Publishers, San Francisco, California, 1998.

[73] Z. Rosberg, H. L. Vu, and M. Zukerman. Performance evaluation of optical burst switching networks with limited wavelength conversion. *Proceedings of ONDM 2003, The 7th IFIP Working Conference on Optical Network Design and Modelling*, pages 1155–1169, Feb. 2003.

[74] Z. Rosberg, H. L. Vu, M. Zukerman, and J. white. Performance analyses of optical burst-switcing networks. *IEEE Journal on Selected Areas in Communications*, 21(7):1187–1197, September 2003.

[75] Sheldon Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, Inc., 1982.

[76] S. Sahu, D. Towsley, and J. Kurose. A quantitative study of differentiated services for the internet. *CMPSCI Technical Report 99-09, University of Massachusetts, MA*, 1999.

[77] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queuing: Achieving approximately fair bandwidth alloallocations in high speed networks. *Proceedings of ACM Sigcomm*, pages 118–130, Oct. 1998.

[78] I. Stoica and H. Zhang. LIRA: An approach for service differentiation in the internet. *Proceedings of Nossdav*, Jun. 1998.

[79] I. Stoika, S. Shenker, and H. Zhang. Core-stateless fair queueing: Achieving approximately fair bandwidth allocation in high speed networks. *Proc. ACM SIGCOMM*, Sep. 1998.

[80] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z-L. Zhang. Exploring the performance benefits of end-to-end path switching. In *Proceedings of ACM SIGMETRICS 2004*, pages 418–419, New York, NY, USA, 2004.

[81] J. Teng and G. N. Rouskas. A comparison of the JIT, JET, and Horizon wavelength reservation schemes on a single OBS node. In *Proceeding of the First International Workshop on Optical Burst Switching*, October 2003.

[82] J. Teng and G. N. Rouskas. On wavelength assignment in optical burst switched networks. In *Proceeding of BROADNETS 2004*, pages 24–33, October 2004.

[83] J. Teng and G. N. Rouskas. Routing path optimization in optical burst switched networks. In *Proceedings of the 9th Conference on Optical Network Design and Modeling (ONDM 2005)*, pages 1–10, February 2005.

[84] Jing Teng. *A Study of Optical Burst Switched Networks with the Jumpstart Just-In-Time Signaling Protocol.* PhD thesis, North Carolina State University, Raleigh, NC, August 2004.

[85] G.P.V. Thodime, V.M. Vokkarane, and J.P. Jue. Dynamic congestion-based load balanced routing in optical burst-switched networks. In *Proceedings of Globecom 2003*, volume 5, pages 2628–2632, December 2003.

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

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

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

[89] V. M. Vokkarane, K. Haridoss, and J. P. Jue. Threshold-based burst assembly policies for QoS support in optical burst-switched networks. *Proceedings of SPIE Optical Networking and Communication Conference (OptiComm)*, 4874:125–136, July-Aug. 2002.

[90] V. M. Vokkarane and J. P. Jue. Prioritized routing and burst segmentation for QoS in optical burst-switched networks. *Proceedings of OFC*, Mar. 2002.

[91] V. M. Vokkarane, Q. Zhang, J. P. Jue, and B. Chen. Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks. *Proceeding of GLOBECOM*, 2002.

[92] S. Y. Wang. Using TCP congestion control to improve the performances of optical burst switched networks. In *IEEE International Conference on Communications (ICC) 2003*, volume 2, pages 1438–1442, May 2003.

[93] X. Wang, H. Morikawa, and T. Aoyama. Burst optical deflection routing protocol for wavelength routing WDM networks. In *Proceedings of Opticomm*, pages 120–129, 2003.

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

[95] J. Y. Wei and R. I. McFarland. Just-in-time signaling for WDM optical burst switching networks. *IEEE/OSA Journal of Lightwave Technology*, 18(12):2019–2037, December 2000.

[96] J. Y. Wei, J. L. Pastor, R. S. Ramamurthy, and Y. Tsai. Just-in-time optical burst switching for multiwavelength networks. In *IFIP TC6 WG6.2 Fifth International Conference on Broadband Communications*, pages 339–352. Kluwer Academic Publishers, November 1999.

[97] P. P. White. RSVP and integrated services in the internet: A tutorial. *IEEE Commun. Mag.*, pages 100–6, May 1997.

[98] J. Widmer, R. Denda, and M. Mauve. A survey on TCP-friendly congestion control. In *IEEE Network*, May/June 2001.

[99] Y. Xiong, M. Vandenhoute, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1838–1851, October 2000.

[100] J. Xu, C. Qiao, J. Li, and G. Xu. Efficient burst scheduling algorithms in optical burst-switched networks using geometric techniques. *IEEE Journal on Selected Areas in Communications*, 22(9):1796–1811, November 2004.

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

[102] L. Xu, H. G. Perros, and G. N. Rouskas. A queueing network model of an edge optical burst switching node. In *Proceedings of IEEE INFOCOM 2003*, pages 2019–2029, April 2003.

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

[104] L. Yang and G. N. Rouskas. A framework for absolute QoS guarantees in optical burst switched networks. *submitted*, 2005.

[105] M. Yang, S. Q. Zheng, and D. Verchere. A QoS supporting scheduling algorithm for optical burst switching DWDM networks. In *Global Telecommunications Conference*, volume 1, pages 86–91, San Antonio, Texas, November 2001.

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

[107] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit. A unified study of contention-resolution schemes in optical packet-switched networks. *Journal of Lightwave Technology*, 21(3):672–683, March 2003.

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

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

[110] X. Yu, C. Qiao, and Y. Liu. TCP implementation and false time out detection in OBS networks. *Proceedings of INFOCOM*, 2004.

[111] X. Yu and Y. Chen C. Qiao. A study of traffic statistics of assembled burst traffic in optical burst switched networks. In *Proceedings of Opticomm*, pages 149–159, Denver, Colorado, August 2001.

[112] A. Zalesky, H. L. Vu, Z. Rosberg, E. M. Wong, and M. Zukerman. Reduced load Erlang fixed point analysis of optical burst switched networks with deflection routing and wavelength reservation. In *Proceedings of the First International Workshop on Optical Burst Switching*, October 2003.

[113] Q. Zhang, V. Vokkarane, J. Jue, and B. Chen. Absolute QoS differentiation in optical burst-switched networks. *IEEE Journal on Selected Areas in Communications*, 22(9):1781–1795, November 2004.