

Generalized Wavelength Sharing Policies for Absolute QoS Guarantees in OBS Networks

Li Yang and George N. Rouskas, *Senior Member, IEEE*

Abstract—We consider the problem of supporting absolute QoS guarantees in terms of the end-to-end burst loss in OBS networks. We present a parameterized model for wavelength sharing which provides for isolation among different traffic classes while also making efficient use of wavelength capacity through statistical multiplexing. We develop a heuristic to optimize the policy parameters for a single link of an OBS network. We also develop 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 in meeting the QoS requirements and keeping the loss rate of best-effort and overall traffic low.

Index Terms—Optical burst switching, wavelength division multiplexing, resource sharing policies, quality of service.

I. INTRODUCTION

OPTICAL burst switching (OBS) [13] 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 OEO conversion at intermediate nodes. OBS transport is positioned between wavelength routing and optical packet switching. The transmission of each burst is preceded by the transmission of a setup message [1], 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.

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. In general, there are two approaches to providing QoS guarantees [15]. 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. Most of the recent research in this area has focused on relative service differentiation, and a variety of schemes have been proposed, such as assigning an additional offset to higher priority bursts [14], intentionally dropping non-compliant bursts [2], and allowing in-profile bursts to preempt out-of-profile ones [10]. A study of absolute QoS guarantees in OBS networks can be found in [15], where two mechanisms were proposed to enforce a loss probability threshold for guaranteed traffic while reducing the loss rate of non-guaranteed traffic: an early dropping mechanism to selectively drop non-guaranteed traffic, and a wavelength grouping strategy to allocate wavelengths to priority traffic. Finally, the study in [9] differs from the above in that it considers delay, rather than burst drop probability, as the QoS parameter to be guaranteed.

In this paper we develop a general framework for absolute service guarantees to users of an OBS network in terms of the end-to-end burst loss. Inspired by earlier work on resource sharing [5], [6], 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 develop a heuristic for optimizing the policy parameters to support per-link absolute QoS guarantees for a given set of heterogeneous traffic classes. Finally, we develop 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 in meeting the QoS requirements and keeping the loss rate of best-effort traffic low.

The paper is organized as follows. In Section II, we discuss the assumptions regarding the OBS network we consider in this study. In Section III we present a suite of parameterized wavelength sharing policies, and in Section IV we develop an algorithm for optimizing the policy parameters for a single OBS link. In Section V 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 Section VI, and we conclude the paper in Section VII.

II. THE OBS NETWORK UNDER STUDY

We consider an OBS network with N nodes. Each link in the network can carry burst traffic on any wavelength

Manuscript received May 7, 2006; revised December 6, 2006. This material is based upon work supported in part by MCNC-RDI under the Jumpstart project and by the NSF under grant ANI-0322107.

L. Yang was with the Department of Computer Science, North Carolina State University. She is now with Scalable Network Technologies, Inc., Los Angeles, CA 90045 (e-mail: lyang@scalable-networks.com).

G. N. Rouskas is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, USA (e-mail: rouskas@ncsu.edu).
Digital Object Identifier 10.1109/TWC.2007.026706.

from a fixed set of W wavelengths, $\{\lambda_1, \lambda_2, \dots, \lambda_W\}$. 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 $i, i = 1, \dots, P-1$, 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:

$$B_i^{e2e} < B_j^{e2e}, \quad 1 \leq i < j \leq P \quad (1)$$

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 $i, i = 1, \dots, P-1$, 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.

III. 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 [6], and in the more general context of resource sharing [5]. 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:

$$W_i^{min} = 0, \quad W_i^{max} = W, \quad i = 1, \dots, P \quad (2)$$

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 a broad class 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 Section II is the subject of Section IV. The analytical models are derived based on the assumption that traffic class $i, i = 1, \dots, P$, 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.

A. The Wavelength Partitioning (WP) Policy

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:

$$0 < W_i^{\min} = W_i^{\max} = W_i < W, \quad i = 1, \dots, P \quad (3)$$

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

$$\sum_{i=1}^P W_i = W \quad (4)$$

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 (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 [15], 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:

$$B_i = \frac{\rho_i^{W_i} / W_i!}{\sum_{j=0}^{W_i} \rho_i^j / j!} \quad (5)$$

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 [6], the performance of complete partitioning can be improved if some sharing of resources is introduced. Next, we describe a class of policies which provide different levels of wavelength sharing among the various traffic classes.

B. The Class of Generalized Wavelength Sharing (GWS) Policies

A policy within the family of generalized wavelength sharing (GWS) policies 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} . The wavelength lower and upper bounds for each class are defined as:

$$0 \leq W_i^{\min} < W_i^{\max} \leq W, \quad i = 1, \dots, P. \quad (6)$$

To allow for wavelength sharing, the sum of wavelength lower bounds over all traffic classes must be less than the total number of wavelengths W , while the sum of the wavelength upper bounds must exceed W ; in other words, the following

constraints are imposed on the wavelength lower and upper bounds:

$$\sum_{i=1}^P W_i^{\min} < W \quad (7)$$

and

$$\sum_{i=1}^P W_i^{\max} > W. \quad (8)$$

More formally, the operation of a GWS policy can be described as follows:

when a class- i burst arrives to find the link at state $P(\underline{n}) = (n_1, \dots, n_P)$, 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:

$$n_i < \min \left\{ W_i^{\max}, W - \sum_{k \neq i} \max \{ n_k, W_k^{\min} \} \right\} \quad (9)$$

Otherwise, the burst is dropped.

To obtain the burst drop probability under the GWS policy, we observe that the state of the OBS link can be described by the vector $\underline{n} = (n_1, \dots, n_P)$, 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:

$$S = \left\{ \underline{n} \mid 0 \leq \sum_{i=1}^P \max \{ W_i^{\min}, n_i \} \leq W, \right. \\ \left. 0 \leq n_i \leq W_i^{\max}, i = 1, \dots, P \right\} \quad (10)$$

Then, the steady state probability of the truncated process has the following product form [8]:

$$P(\underline{n}) = P(n_1, n_2, \dots, n_P) = \begin{cases} C \frac{\rho_1^{n_1} \rho_2^{n_2} \dots \rho_P^{n_P}}{n_1! n_2! \dots n_P!}, & \underline{n} \in S \\ 0, & \underline{n} \notin S \end{cases} \quad (11)$$

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:

$$C^{-1} = \sum_{\underline{n} \in S} \frac{\rho_1^{n_1} \rho_2^{n_2} \dots \rho_P^{n_P}}{n_1! n_2! \dots n_P!} \quad (12)$$

Since the cardinality of the state space S is almost $O(W^P)$ [7], 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 [3]. This algorithm is based on the numerical inversion approach introduced in [4]. In this work, we adopt the direct method in [3], which is appropriate for the system sizes we consider, and we calculate the normalizing constant via the P -fold nested sum:

$$C^{-1} = \sum_{n_1=0}^{K_1} \frac{\rho_1^{n_1}}{n_1!} \sum_{n_2=0}^{K_2} \frac{\rho_2^{n_2}}{n_2!} \dots \sum_{n_P=0}^{K_P} \frac{\rho_P^{n_P}}{n_P!} \quad (13)$$

The P upper limits K_i , $i = 1, \dots, P$, in the above summation are not independent; their value depends on both the wavelength upper and lower bounds W_i^{max} and W_i^{min} , respectively, and the values of parameters n_k , $k = 1, \dots, i-1$, in expression (13), as follows:

$$K_i = \min \left\{ W_i^{max}, W - \sum_{k=1}^{i-1} \max \{ n_k, W_k^{min} \} - \sum_{k=i+1}^P W_k^{min} \right\} \quad (14)$$

Note that the above expression is slightly different than (9). In the latter expression, the values of all parameters n_i are considered fixed at the time of a burst arrival that finds the system at state $P(\underline{n}) = P(n_1, n_2, \dots, n_P)$. On the other hand, when enumerating all terms of the nested sum (13) for variable n_i , only the values of parameters n_1, \dots, n_{i-1} are considered fixed; hence, the value of n_i can be constrained only by the lower bounds W_k^{min} of parameters n_{i+1}, \dots, n_P , as shown in the rightmost term of expression (14).

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 $C^{-1}(W, \underline{W}^{min}, \underline{W}^{max})$ to 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. 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:

$$B_i = 1 - \frac{C^{-1}(W-1, \underline{W}^{min} - \underline{1}_i, \underline{W}^{max} - \underline{1}_i)}{C^{-1}(W, \underline{W}^{min}, \underline{W}^{max})} \quad (15)$$

Due to the Poisson arrival assumption, expression (15) also represents the probability that an arriving class- i burst will be dropped.

IV. 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 , $i = 1, \dots, P$, as a function of the offered loads ρ_i and the wavelength lower and 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.

A. The Policy Optimization Problem

As we discussed earlier, we consider an OBS link with W wavelengths and P classes of traffic. Each traffic class i , $i = 1, \dots, P$, 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 $j > i$:

$$B_i^\ell < B_j^\ell, \quad 1 \leq i < j \leq P \quad (16)$$

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 . We emphasize that the use of the Erlang-B formula for the burst drop probability is only an approximation, albeit one commonly used in the literature. As it was pointed out in [15], each guaranteed class i must be allocated W_i wavelengths such that:

$$W_i = Erl^{-1}(\rho_i, B_i^\ell), \quad i = 1, \dots, P-1 \quad (17)$$

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 generalized wavelength sharing (GWS) policies.

Let us assume that the OBS link operates under a GWS policy. Let ρ_i and B_i^ℓ be the offered load and link loss guarantee, respectively, of traffic class i , $i = 1, \dots, P$ (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 , $i = 1, \dots, P-1$ below B_i^ℓ . More formally, this optimization problem can be stated as:

minimize: B_P

subject to:

$$B_i \leq B_i^\ell, \quad i = 1, \dots, P-1 \quad (18)$$

$$0 \leq W_i^{min} \leq W_i^{max} \leq W, \quad i = 1, \dots, P \quad (19)$$

$$W_i^{min}, W_i^{max} : \text{integer}, \quad i = 1, \dots, P \quad (20)$$

and the constraints (7) and (8); the burst drop probabilities B_i , $i = 1, \dots, P$, are obtained from expression (15).

Clearly, the above is an integer optimization problem with a nonlinear objective function and nonlinear constraints (18). Furthermore, important mathematical properties such as monotonicity and convexity have not been established for this type of objective function [5]. 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.

B. 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 $i, i = 1, \dots, P - 1$. 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 $i, i = 1, \dots, P$, at the end of iteration $k, k = 0, 1, 2, \dots$. 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 (15). 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 $i, i = 1, \dots, P - 1$. 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 (18) 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 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 (21), shown at the top of the next 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.

GWS Policy Optimization for an OBS Link

Input: An OBS link with W wavelengths, P traffic classes, offered load ρ_i and burst loss guarantee $B_i^\ell, i = 1, \dots, P$ ($B_P^\ell = 1.0$)

Output: Pair of wavelength lower and upper bounds $(W_i^{min}, W_i^{max}), i = 1, \dots, P$, such that $B_i \leq B_i^\ell, i = 1, \dots, P - 1$, and B_P is minimized

procedure **PolicyOpt**

begin

1. $k \leftarrow 0$ // iteration index
 2. **for** $i = 1$ to $P - 1$ **do** // initialization
 3. $W_i^{min}(k) \leftarrow Erl^{-1}(\rho_i, B_i^\ell)$;
 4. $W_i^{max}(k) \leftarrow \min\{2W_i^{min}(k), W\}$
 5. $(W_P^{min}(k), W_P^{max}(k)) \leftarrow$ pair of values that minimizes $B_P(0)$ without violating constraints (18)
 6. **repeat** // main iteration
 7. $k \leftarrow k + 1$
 8. Let m be the class with the minimum value of $\frac{B_i(k-1)}{B_i^\ell}, i = 1, \dots, P - 1$
 9. $\mathcal{L}(k) \leftarrow$ the local neighborhood from expression (21)
 10. $B \leftarrow 1.0$ // temporary variable
 - // update the wavelength bounds of classes m and P
 11. **for each** $(w_m^{min}, w_m^{max}) \in \mathcal{L}(k)$ **do**
 12. $(w_P^{min}, w_P^{max}) \leftarrow$ pair of values that minimizes B_P without violating constraints (18)
 13. **if** $B_P < B$ **then**
 14. $W_m^{min}(k + 1) \leftarrow w_m^{min}, W_m^{max}(k + 1) \leftarrow w_m^{max}$
 $W_P^{min}(k + 1) \leftarrow w_P^{min}, W_P^{max}(k + 1) \leftarrow w_P^{max}$
 // wavelength bounds of other classes remain the same
 15. **for** $i = 1$ to $P - 1, i \neq m$ **do**
 16. $W_i^{min}(k + 1) \leftarrow W_i^{min}(k)$
 $W_i^{max}(k + 1) \leftarrow W_i^{max}(k)$
 17. **until** B_P cannot be decreased any further
 18. **if** $\sum_{i=1}^{P-1} W_i^{min} \geq W$ **then**
 19. **return error** // cannot meet QoS
 20. **else return** $(W_i^{min}(k), W_i^{max}(k)), i = 1, \dots, P$
- end**

Fig. 1. Local search heuristic for GWS policy optimization

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 (15) 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.

$$\mathcal{L}(k+1) = \begin{cases} \{(W_m^{min}(k), W_m^{max}(k) - 1), (W_m^{min}(k) + 1, W_m^{max}(k) - 1)\}, & W_m^{min}(k) = 0 \\ \{(W_m^{min}(k) - 1, W_m^{max}(k) - 1), (W_m^{min}(k) - 1, W_m^{max}(k)), \\ (W_m^{min}(k), W_m^{max}(k) - 1), (W_m^{min}(k) + 1, W_m^{max}(k) - 1), \\ (W_m^{min}(k) - 1, W_m^{max}(k) + 1)\}, & W_m^{min}(k) > 0 \end{cases} \quad (21)$$

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 $B_i^\ell, i = 1, \dots, P-1$, 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 (17)) for guaranteed class i . At the beginning of the algorithm, for the guaranteed classes we let:

$$W_i^{min}(0) = W_i, W_i^{max}(0) = \min\{2W_i, W\}, i = 1, \dots, P-1 \quad (22)$$

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 (18).

A step-by-step description of the local search algorithm is provided in Figure 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 (15) with the new bounds. Nevertheless, our experimental results indicate that our algorithm converges to a local optimum after only a few iterations.

V. 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 a generalized wavelength sharing (GWS) 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:

$$B_1^{e2e} < B_2^{e2e} < \dots < B_{P-1}^{e2e} < B_P^{e2e} = 1.0 \quad (23)$$

The main issue we address in this section is how to optimize the parameters of the GWS 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 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 $B_i^{e2e}, i = 1, \dots, P-1$. 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:

$$B_i^\ell(h) = 1 - \exp\left(\frac{\ln(1 - B_i^{e2e})}{h}\right), i = 1, \dots, P-1 \quad (24)$$

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 (15) and the corresponding increase in the running time of the policy optimization algorithm.

A simple solution to this problem was suggested in [15], 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 (24). 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. Conse-

quently, 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 [15]. 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, as suggested in [15]. 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 (24). 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 , $i = 1, \dots, P$ is sufficient to meet the end-to-end QoS. To this end, we compute the network-wide end-to-end burst loss probability of class- i traffic as [11]:

$$\mathcal{B}_i = \frac{\sum_{l \in E} B_i^\ell \times \rho_i^{(l)}}{\sum_{s,d} \rho_i^{(s,d)}}, \quad i = 1, \dots, P-1 \quad (25)$$

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 we let B_i^ℓ , $i = 1, \dots, P$, be the midpoint $B_i^{mid} = (B_i^{min} + B_i^{max})/2$ 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 (25), 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 $B_i^\ell = B_i^{min}$, $i = 1, \dots, P-1$.

The details of this binary search algorithm can be found in Figure 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.

Let us now discuss the implementation aspects of the GWS policy. Deployment of GWS involves two phases. In the off-line phase, the upper and lower wavelength bounds W_i^{min} and W_i^{max} , respectively, for each traffic class i , $i = 1, \dots, P$ are computed for a given network topology, traffic pattern,

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 $\underline{B}^{e2e} = (B_1^{e2e}, \dots, B_{P-1}^{e2e})$

Output: Per-link loss guarantee vector $\underline{B}^\ell = (B_1^\ell, \dots, B_{P-1}^\ell)$ 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 (24)

1. $\underline{B}^{min} \leftarrow (B_1^\ell(D), \dots, B_{P-1}^\ell(D))$

2. $\underline{B}^{max} \leftarrow (B_1^\ell(\bar{H}), \dots, B_{P-1}^\ell(\bar{H}))$

2. **while** $\underline{B}^{max} > \underline{B}^{min} \times \epsilon$ **do** // binary search

3. $\underline{B}^{mid} \leftarrow (\underline{B}^{min} + \underline{B}^{max}) / 2$

4. $\underline{\mathcal{B}} \leftarrow (\mathcal{B}_1, \dots, \mathcal{B}_{P-1})$ from (25) with $\underline{B}^\ell = \underline{B}^{min}$

// attempt to increase the link guarantees to decrease B_P

5. **if** $\underline{\mathcal{B}} < \underline{B}^{e2e}$ **then**

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

Fig. 2. Binary search algorithm for selecting the per-link loss guarantees

and end-to-end guarantees, using the techniques we described in this work; we expect this computation to take place in a centralized manner at a network control or management node. Since we assume that the link loss guarantee of each traffic class is the same on each link, the values of W_i^{min} and W_i^{max} only depend on the class i and are the same for all links. Hence, the network links can be configured appropriately by simply downloading this set of values to each network node. We also expect that the control node will monitor the traffic conditions and network topology and update the values of W_i^{min} and W_i^{max} periodically to reflect any changes; however, any such updates will take place at time scales much longer (i.e., a few minutes or hours) compared to burst transmissions. During the on-line phase, network nodes use the configured values of W_i^{min} and W_i^{max} to determine whether an arriving burst can be accepted or not. This operation only requires the nodes to keep track of the number of bursts from each class currently served by the link. In particular, there is no need to maintain any online traffic statistics, and the decision to accept or reject a burst depends only on its traffic class independently of the state of other classes. Hence, the GWS policy is simple to implement in hardware and can be readily deployed for further evaluation in a testbed environment.

VI. NUMERICAL RESULTS

A. 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

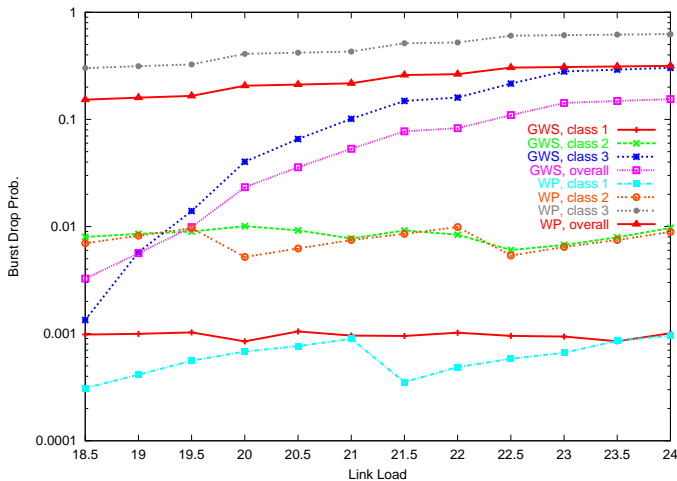


Fig. 3. Single link with $W = 32$ wavelengths and $P = 3$ traffic classes, $\rho_1 = 0.2\rho$, $\rho_2 = 0.3\rho$, $\rho_3 = 0.5\rho$.

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 Section III-A and also considered in [15], 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 (17).
- 2) The GWS policy, described in Section III-B, which associates a pair of wavelength 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 1.

Figure 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 GWS; 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 GWS 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 GWS policy is one-half that under the WP policy. More importantly, the GWS policy reduces the overall burst drop rate significantly, with a corresponding substantial increase in throughput.

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

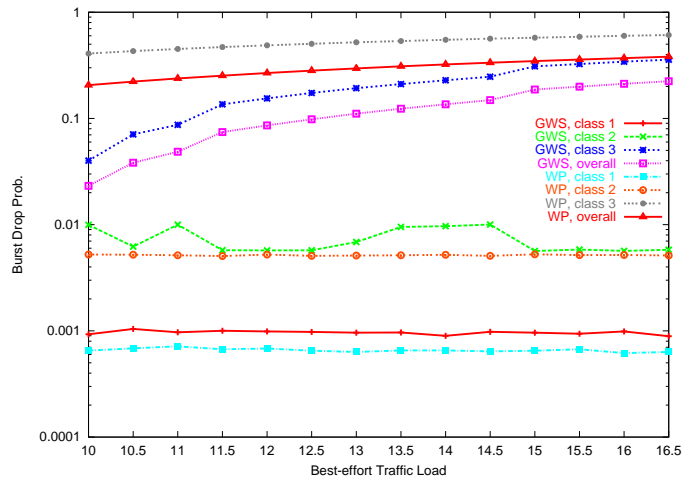


Fig. 4. Single link with $W = 32$ wavelengths and $P = 3$ traffic classes, $\rho_1 = 4$ Erlang, $\rho_2 = 6$ Erlang.

link capacity into three sets of wavelengths, each dedicated to carrying bursts in one of the three traffic classes. The GWS 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 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 GWS 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 over-provisions the guaranteed classes. This is evident from the behavior of the burst loss curves for the guaranteed classes under the WP policy in Figure 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 GWS 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 GWS 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, 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 GWS

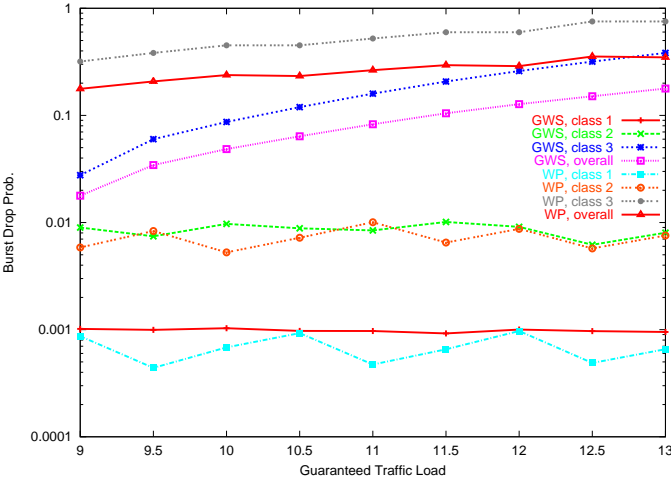


Fig. 5. Single link with $W = 32$ wavelengths and $P = 3$ traffic classes, $\rho_3 = 11$ Erlang, $\rho_2 = 1.5\rho_1$.

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 GWS 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 GWS 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 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 GWS 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 GWS policy.

B. 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. We use the simulator that was developed as part of the Jumpstart project [12]. The simulator accounts for all the details of the Jumpstart OBS signaling protocol [1] which employs the Just-In-Time (JIT) reservation scheme. (We emphasize, however, that the wavelength sharing policies we present and evaluate in this work are independent of the specifics of the reservation protocol, and can be deployed alongside either the JET or the Horizon reservation schemes.) 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;

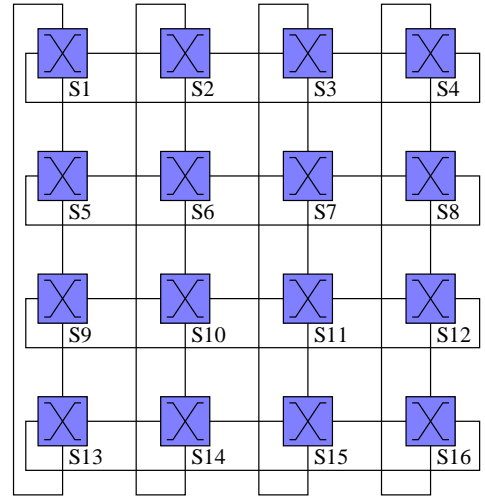


Fig. 6. The 4×4 torus network.

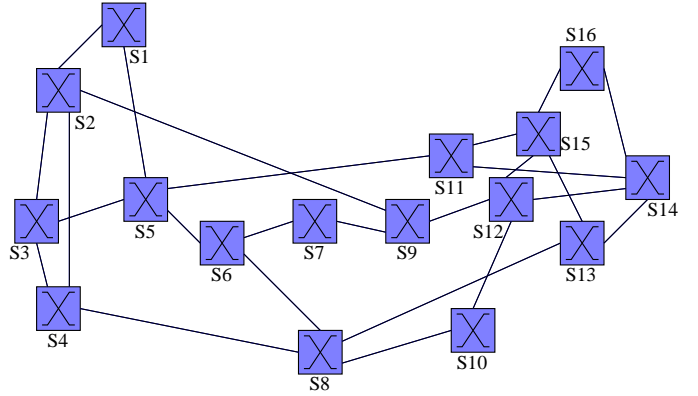


Fig. 7. The 16-node topology based on the 14-node NSFNet.

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 6 is based on a regular topology, while the network in Figure 7 is based on an irregular topology derived from the 14-node NSF network. 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 1, is $\bar{H}_{NSF} = 2.283$ and $\bar{H}_{torus} = 2.133$.

In Figure 8, we plot the overall burst drop probability, as well as that of the three classes of traffic, under the two

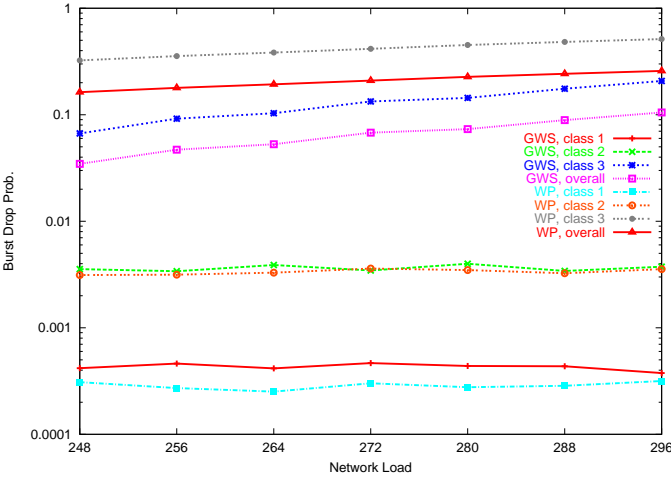


Fig. 8. NSFNet, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained from (24) with $h = D$.

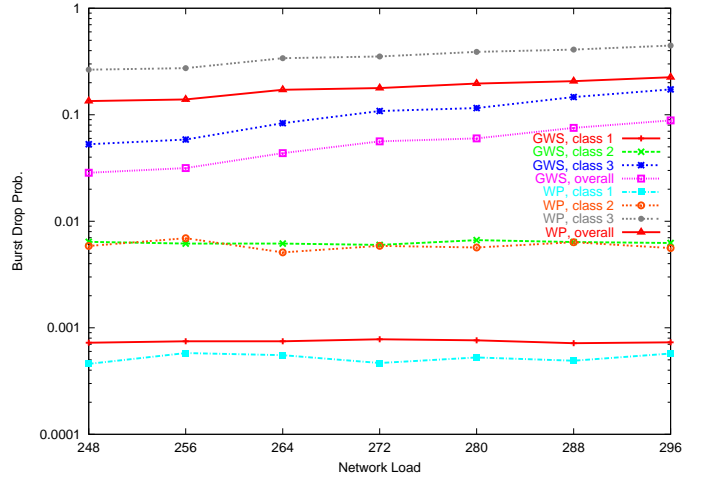


Fig. 10. NSFNet, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained by the optimization procedure in Figure 1.

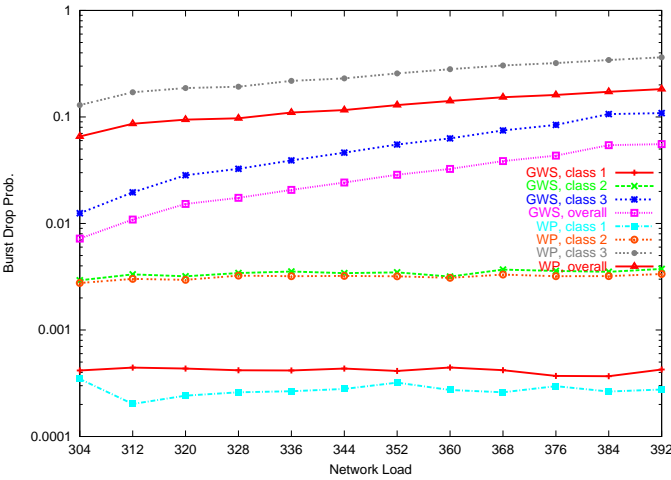


Fig. 9. Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained from (24) with $h = D$.

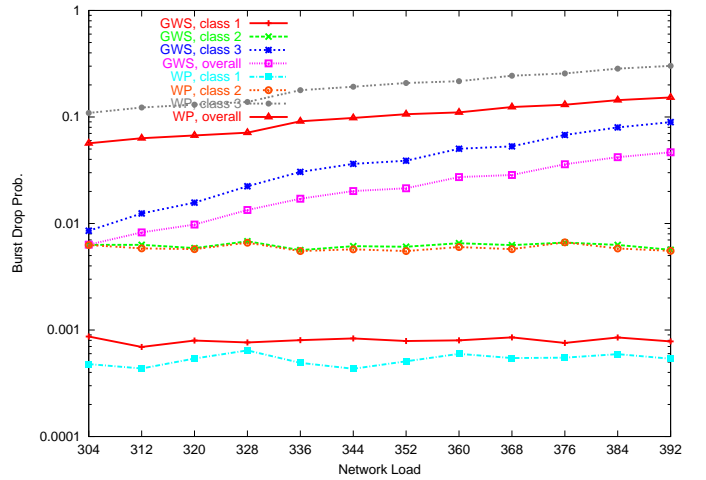


Fig. 11. Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, uniform pattern, B_i^ℓ obtained by the optimization procedure in Figure 1.

policies, WP and GWS, 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 (24); this is the approach suggested in [15]. Figure 9 shows similar results for the torus network. Our observations regarding the relative behavior of the two policies, WP and GWS, 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 GWS 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 over-provisioning 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 over-provisioning problem, we used the optimization procedure in Figure 1 to determine an appropriate value for the link-loss guarantee B_i^ℓ , $i = 1, 2$, given the corresponding end-to-end loss guarantee B_i^{e2e} . The simulation results are shown in Figures 10 and 11, for the NSFNet and

torus networks, respectively. Comparing to Figures 8 and 9, 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.

In Figures 12 and 13 we present results for the torus topology with the distance-dependent traffic pattern; for the former figure, the link-loss guarantees were obtained from (24) with $h = D = 4$, while for the latter, they were obtained by the optimization procedure in Figure 1. Similar results were obtained for the NSFNet. 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.

As a final note, Figures 8-13 plot the network-wide drop probability for each class, which is an average of the drop probabilities along all paths, including short and long ones. Since the results in Figures 8, 9, and 12, were obtained by

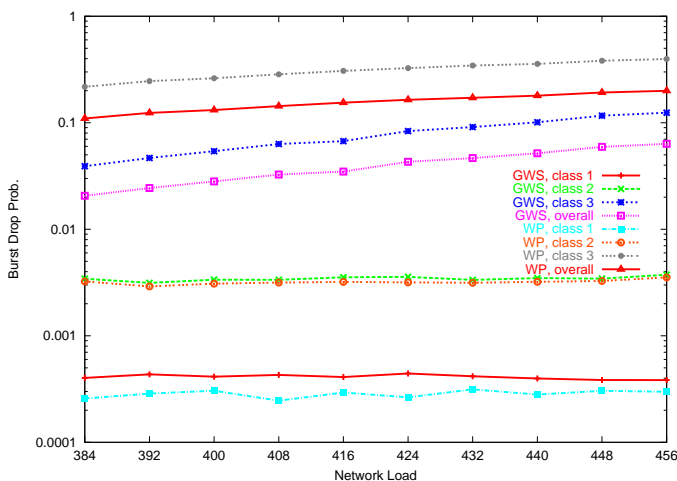


Fig. 12. Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, distance-dependent pattern, B_i^ℓ obtained from (24) with $h = D$.

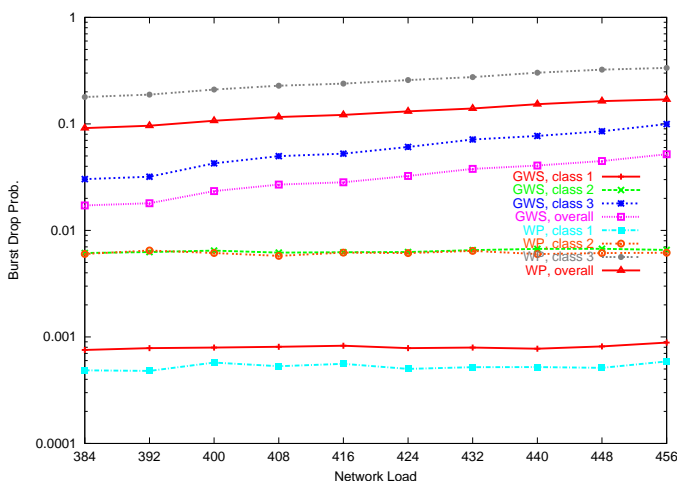


Fig. 13. Torus, $W = 32$ wavelengths, $P = 3$ traffic classes, B_i^ℓ obtained by the optimization procedure in Figure 1.

setting the per-link loss guarantees based on the network diameter D , the end-to-end guarantees are met over all paths in the network. On the other hand, the results in Figures 10, 11, and 13, were obtained by optimizing the per-link guarantees with a value $h < D$. Although the network-wide drop probability of the overall traffic is lower in this case, the end-to-end guarantees may not be met for paths of length greater than h . Therefore, for classes of traffic for which guaranteeing the end-to-end drop probability over the longest path is a strict requirement, the link loss guarantee must be set based on the network diameter D . However, for classes of traffic which can cope with a slightly higher drop probability over the longest path (e.g., when the cost of recovering at a higher layer packets carried by a dropped burst is relatively low), using our approach to optimize the per-link guarantees will result in an increase of the network-wide traffic carrying capacity due to lower burst drop rates.

VII. CONCLUDING REMARKS

We have presented 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 generalized

wavelength sharing (GWS) approach is effective and efficient in managing the wavelength resources, is simple to implement, and outperforms previously proposed methods. The GWS scheme may also be combined with other mechanisms such as early burst drop or path clustering [15] to further improve its performance.

REFERENCES

- [1] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "JumpStart: a just-in-time signaling architecture for WDM burst-switched networks," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 82–89, Feb. 2002.
- [2] Y. Chen, M. Hamdi, and D. H. K. Tsang, "Proportional QoS over OBS network," in *Proc. IEEE GLOBECOM 2001*, vol. 3, pp. 1510–1514.
- [3] 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*, vol. 27, pp. 1104–1143, 1995.
- [4] G. L. Choudhury, K. K. Leung, and W. Whitt, "Calculating normalization constants of closed queueing networks by numerically inverting their generating functions," *J. of the ACM*, 1993.
- [5] G. L. Choudhury, K. K. Leung, and W. Whitt, "Efficiently providing multiple grades of service with protection against overloads in shared resources," *AT&T Techn. J.*, vol. 74, pp. 50–63, 1995.
- [6] F. Kamoun and L. Kleinrock, "Analysis of shared finite storage in a computer network node environment under general traffic conditions," *IEEE Trans. Commun.*, vol. 28, no. 7, July 1980.
- [7] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. 29, no. 10, pp. 1474–1481, Oct. 1981.
- [8] F. P. Kelly, *Reversibility and Stochastic Networks*. New York: John Wiley & Sons, 1979.
- [9] J. Liu and N. Ansari, "Forward resource reservation for QoS provisioning in OBS systems," in *Proc. IEEE GLOBECOM 2002*.
- [10] C.-H. Loi, W. Liao, and D.-N. Yang, "Service differentiation in optical burst switched networks," in *Proc. IEEE GLOBECOM 2002*, vol. 3, pp. 2313–2317.
- [11] J. Teng and G. N. Rouskas, "Routing path optimization in optical burst switched networks," in *Proc. 9th Conference on Optical Network Design and Modeling (ONDM 2005)*, pp. 1–10.
- [12] J. Teng, *A study of optical burst switched networks with the Jumpstart Just-In-Time signaling protocol*, Ph.D. thesis, North Carolina State University, Raleigh, NC, Aug. 2004.
- [13] L. Xu, H. G. Perros, and G. N. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Commun. Mag.*, vol. 39, no. 1, pp. 136–142, Jan. 2001.
- [14] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," *J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 2062–2071, Oct. 2000.
- [15] Q. Zhang, V. Vokkarane, J. Jue, and B. Chen, "Absolute QoS differentiation in optical burst-switched networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 9, pp. 1781–1795, Nov. 2004.



Li Yang earned a PhD in Computer Science from North Carolina State University in 2006, an MS in Computer Science from Beijing University of Posts and Telecoms (BUPT) in 2000, and a BS in Computer Engineering from Xi'an Institute of Posts and Telecoms in 1997. She worked for a year as a software engineer in Alcatel Shanghai Bell-BUPT Research and Development Center in Beijing, China. She also worked as a software intern at Intel Raleigh Design Center. Dr. Yang joined Scalable Networks as a Research Scientist in 2006.



George N. Rouskas (S '92, M '95, SM '01) is a Professor of Computer Science at North Carolina State University. He received the Diploma in Computer Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. During the 2000-2001 academic year he spent a sabbatical term at Vitesse Semiconductor, Morrisville, NC, and in May 2000, December 2002,

and July 2006 he was an Invited Professor at the University of Evry, France. His research interests include network architectures and protocols, optical networks, multicast communication, and performance evaluation.

Dr. Rouskas received the 2004 ALCOA Foundation Engineering Research Achievement Award, and the 2003 NCSU Alumni Outstanding Research Award. He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award, a co-author of a paper that received the Best Paper Award

at the 1998 SPIE conference on All-Optical Networking, and the recipient of the 1994 Graduate Research Assistant Award from the College of Computing, Georgia Tech. Dr. Rouskas is especially proud of his teaching awards, including his induction in the NCSU Academy of Outstanding Teachers in 2004, and the Outstanding New Teacher Award he received from the Department of Computer Science in 1995.

Dr. Rouskas is founding co-editor-in-chief of *Optical Switching and Networking (OSN)*, an Elsevier journal, he has served on the editorial boards of the *IEEE/ACM Transactions on Networking*, *Computer Networks*, and *Optical Networks*, and he was a co-guest editor for the *IEEE Journal on Selected Areas in Communications*, Special Issue on Protocols and Architectures for Next Generation Optical WDM Networks, published in October, 2000. He was technical program co-chair of the Networking 2004 conference, general co-chair of the IEEE LANMAN 2005 workshop, program chair of the IEEE LANMAN 2004 workshop, and program co-chair of the Traffic Grooming workshop 2004. He is serving as general co-chair for BROADNETS 2007. He is a member of the ACM and of the Technical Chamber of Greece.