Traffic engineering approach to path selection in optical burst switching networks

Jing Teng

Pionic Technology, Raleigh, North Carolina 27606

George N. Rouskas

Department of Computer Science, North Carolina State University, Raleigh, North Carolina 27695-8206

rouskas@csc.ncsu.edu

RECEIVED 10 AUGUST 2005; REVISED 15 SEPTEMBER 2005; ACCEPTED 28 SEPTEMBER 2005; PUBLISHED 27 OCTOBER 2005

It is usually assumed that optical burst switching (OBS) networks use the shortest path routing along with next-hop burst forwarding. The shortest path routing minimizes delay and optimizes utilization of resources, however, it often causes certain links to become congested while others remain underutilized. In a bufferless OBS network in which burst drop probability is the primary metric of interest, the existence of a few highly congested links could lead to unacceptable performance for the entire network. We take a traffic engineering approach to path selection in OBS networks with the objective of balancing the traffic across the network links to reduce congestion and to improve overall performance. We present an approximate integer linear optimization problem as well as a simple integer relaxation heuristic to solve the problem efficiently for large networks. Numerical results demonstrate that our approach is effective in reducing the network-wide burst drop probability, in many cases significantly, over the shortest path routing. © 2005 Optical Society of America

OCIS codes: 060.4250, 060.0600.

1. Introduction

Optical burst switching (OBS) is a promising switching paradigm that 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, and 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 optical packet switching. All-optical circuits tend to be inefficient for traffic that has not been groomed or statistically multiplexed, whereas optical packet switching 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 use 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.

Over the past few years, research in OBS networks has rapidly progressed from purely theoretical investigations to prototypes and proof-of-concept demonstrations. For a recent overview of the breadth and depth of current OBS research, the reader is referred to Ref. [1]. Yet despite the multitude of directions that OBS research has taken, there is one important area, namely, the selection of routing paths, that has received relatively little attention in the literature despite the profound impact that routing can have on the overall performance of an OBS network. In particular, most studies that investigate the performance of OBS networks assume (either implicitly or explicitly) that bursts are routed over the shortest path to their destination. 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 situation 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 could lead to unacceptably high burst loss for the entire network.

One possible routing mechanism that can be used to reduce the burst loss due to suboptimal path selection (e.g., shortest paths) is deflection routing [2]. 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 Ref. [3], whereas Refs. [4, 5] analyzed the performance of deflection routing. However, deflection routing in OBS networks has several disadvantages. A practical implementation would require intermediate switches that 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 that can have disastrous effects in an optical network [6]. Finally, deflection routing is by nature suboptimal since it considers only the congestion of the current switch, not the state of the links further along the path; and it can cause undesirable vibration effects, as explained in Ref. [5].

Here we take a traffic engineering approach to path selection in OBS networks. Our objective is to determine a set of routing paths so as to minimize the overall burst drop probability in the network. The main idea is to balance the burst traffic across the network links to reduce congestion and to improve overall performance. To this end, we develop a traffic flow model and present a linear optimization problem by establishing the relationship between the (approximate) overall burst drop probability and the traffic flow vector. Although the problem formulation is based on certain simplifying assumptions (that we discuss and justify), the paths obtained through the solution to this problem tend to balance the burst traffic evenly, reducing congestion and improving the performance significantly compared with the shortest path routing.

The paper is organized as follows. In Section 2 we discuss our assumptions with regard to the OBS network we consider in our study. In Section 3 we formulate a linear optimization problem with the objective of minimizing the overall burst drop probability, and we show how to solve it to obtain a set of optimal paths. In Section 4 we present simulation results to demonstrate the effectiveness of our approach; we conclude the paper in Section 5.

2. Optical Burst Switching Network under Study

An OBS network is composed of users, optical switches (nodes), and fibers. Users are devices, e.g, high-speed electronic routers or multiplexers, that generate optical bursts. An optical switch consists of two components: an optical cross connect (OXC) that can optically forward a burst from an input to an output port without OEO conversion; and a signaling engine that 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.

We use $G = (V,E)$ to denote an OBS network; $V = \{S_1,S_2,\ldots,S_N\}$ is the set of switches, $N =| V |$; and $E = \{\ell_1,\ell_2,\ldots,\ell_M\}$ is the set of unidirectional fiber links, $M =| E |$. If a link ℓ_k connects an output port of switch S_i to an input port of switch S_j, we will refer to S_i and S_j as the tail and head, respectively, of ℓ_k. We also define $\text{tail}\,(i) = \{\ell_k \mid S_i \text{ is the tail of } \ell_k\}$, as the set of links with S_i as their tail; similarly $\text{head}\,(i) = \{\ell_k \mid S_i \text{ is the head of } \ell_k\}$ is the set of links with S_j as their head. Each link in the network can carry burst traffic on any wavelength from a fixed set of W wavelengths, $\{\lambda_1,\lambda_2,\ldots,\lambda_W\}$.

We assume that the OBS network employs source routing, in that the ingress switch (source) determines the path of a burst that enters the network. The path over which the burst must travel is carried by the setup message that precedes the transmission of the data burst. The network uses either fixed-path or multipath routing. In fixed-path routing, all the bursts between a source–destination pair follow the same path through the network. In multipath routing, a burst can take one of a (small) number of paths to its destination. We assume that the 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.

We also assume that each OBS switch in the network has full wavelength conversion capabilities that 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. 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.

Unlike in circuit-switched (i.e., wavelength routing) networks, where a transmission starts only after an end-to-end path reservation is acknowledged, OBS networks use one-way reservations, and bursts follow the transmission of their respective setup messages without waiting for an acknowledgment that a reservation was successful [7]. Since bursts cannot be buffered inside the network, a burst can successfully travel along several links and still be dropped at a congested switch before reaching its destination. Also, in OBS networks, switch resources are allocated for an amount of time necessary to switch and transmit an individual burst; in circuit switching, on the other hand, the resources are exclusively reserved for the entire duration of the end-to-end connection. As a result, if the link propagation delay is significantly larger than the transmission time of a burst, several bursts from different connections or source–destination pairs could be in flight simultaneously along the same wavelength link.

3. Path Optimization for Optical Burst Switching Networks

We take a traffic engineering approach to compute a set of paths in an OBS network so as to minimize the overall burst drop probability. We assume that the traffic pattern is described by an $N \times N$ matrix $\Gamma = [\gamma_{ij}]$, where γ_{ij} represents the (long-term) arrival rate of bursts that originate at switch S_i and are destined for switch S_j. The values of the traffic elements γ_{ij} can be obtained empirically, or they can be based on predictions with regard to the long-term demands placed on the network; although these values can be updated from time to time, we assume that any such changes in the traffic matrix take place over long time scales and that routing paths remain fixed during the time between successive updates in the traffic matrix. Let $1/\mu_{ij}$ denote the mean length of bursts that travel from switch S_i to switch S_j; we will use $\rho_{ij} = \gamma_{ij}/\mu_{ij}$ to denote the offered load of bursts from S_i to S_j.

Given a demand matrix, a typical approach to determine a set of paths that optimizes a certain performance metric of interest (e.g., congestion and average delay) is to formulate and solve an optimization problem (refer to [8] and references therein for similar problems in wavelength routed networks). We take a similar approach in that we formulate a linear optimization problem to determine the optimal routing paths; in our case, the objective is to minimize the burst drop probability over the entire network, and the demand matrix is determined by the offered load values $\left\{\rho_{ij}, i, j = 1, \ldots, N\right\}$. However, we note that the problem at hand differs from typical network flow problems [9] in several aspects:

- It is impossible to express the objective function (overall burst drop probability) as a function of the link burst drop probabilities in an exact and closed-form manner. (For example, the reduced load fixed point approximation in Ref. [10] can be used to obtain an accurate estimate of the burst drop probability in the network; however, it is an iterative process and does not yield a closed-form expression).
- Even if one were to use an approximate expression for the objective function, the resulting formulation would not be linear.
- The link burst drop probabilities (and, hence, the objective function) depend not on the known quantities ρ_{ij} (the offered load), but rather on the actual loads that are unknown and can be determined only after the optimal paths have been obtained (the actual load on a link that is due to a certain traffic component equals the offered load of that component minus an amount corresponding to the burst traffic dropped at previous links of the component's path).
- The relationship between the link burst drop probabilities and the corresponding link loads depends strongly on the nature of the burst traffic (e.g., Poisson and self-similar); for non-Poisson burst arrival models, this relationship can be difficult (or even impossible) to express analytically.

Next we present a formulation that overcomes the above difficulties and allows us to obtain routing paths that improve the burst drop probability significantly over the shortest-path routing by distributing burst traffic over the network paths so as to reduce link contention. We emphasize that our main goal has been to obtain a practical formulation that can be solved efficiently for large networks. To this end, we have made certain approximations to obtain a linear model and to avoid complex and computationally expensive formulations. The following discussion explains our assumptions and notations.

3.A. Traffic Flow Model Formulation

Our first step is to formulate a traffic flow model for optimization by establishing the relationship between the (approximate) overall burst drop probability and the traffic flow vector. Let $\beta^{(k)}$ denote the probability that a burst is dropped along link ℓ_k of the network. We make the reasonable assumption that $\beta^{(k)} \ll 1, \forall k$, and also that the drop probability along link ℓ_k is independent of the source or destination of a burst, or the path it has followed before entering link ℓ_k. Then, the burst drop probability $b(\pi)$ along a path π is given by

$$b(\pi) = 1 - \prod_{\ell_k \in \pi} \left[1 - \beta^{(k)}\right] \simeq \sum_{\ell_k \in \pi} \beta^{(k)} \ll 1. \tag{1}$$

Therefore, we assume that the actual traffic load $\hat{\rho}_{ij}$ seen by the network that is due to traffic that originates at switch S_i and terminates at switch S_j is equal to the offered load of this traffic component, ρ_{ij} (i.e., there is no traffic loss). Obviously, this is an approximation that is more accurate when the burst drop probability is low, but one that significantly simplifies the formulation.

Let $x_{ij}^{(k)}$ denote the fraction of burst traffic from switch S_i to switch S_j that travels over link ℓ_k, $0 \le x_{ij}^{(k)} \le 1$; quantities $x_{ij}^{(k)}$ constitute the traffic flow vector. Then, the burst drop probability B_N over all the burst traffic in the network is given by

$$B_N = \frac{\sum_{\ell_k \in E} \left[\beta^{(k)} \times \sum_{i \neq j} \rho_{ij} x_{ij}^{(k)}\right]}{\sum_{i \neq j} \rho_{ij}} = \frac{\sum_{\ell_k \in E} \beta^{(k)} \times \rho^{(k)}}{\sum_{i \neq j} \rho_{ij}}, \tag{2}$$

where $\rho^{(k)}$ is the total load seen by link ℓ_k under the assumption that there is no traffic loss.

Given the traffic demands $\{\rho_{ij}\}$, our objective is to minimize the network-wide burst drop probability B_N in Eq. (2). As we mentioned previously, however, the expression for B_N depends on the burst arrival model. In general, it might not be possible to express B_N as a linear function of $x_{ij}^{(k)}$, and in fact, it might be impossible to obtain even a closed-form expression for B_N. To overcome this problem, we make the assumption that the burst arrival process to each link in the network is Poisson. This is clearly an approximation since, even if arrivals to the network are Poisson, burst arrivals to a given link are reduced due to loss in previous links and are not Poisson. However, whenever burst loss is small, we can assume that the thinned process remains Poisson. Furthermore, the Poisson assumption allows us to develop a linear problem formulation from which a set of routing paths can be obtained. Even if the arrival process is not Poisson, the routing paths obtained with our approach help to reduce the burst drop probability (compared to schemes such as shortest-path routing) since they tend to balance the load more evenly among the network links. Finally, our approach can be adapted to non-Poisson traffic if the link drop probabilities in such a case can be approximated by a convex function (as we discuss shortly).

Under the Poisson arrival assumption, the burst drop probability at each link ℓ_k is given by the Erlang-B formula:

$$\beta^{(k)} = Erl\left[\rho^{(k)}, W\right] = \frac{\left[\rho^{(k)}\right]^W}{\sum_{i=0}^{W} \left[\rho^{(k)}\right]^i}, \tag{3}$$

where W is the number of wavelengths at link ℓ_k. Let us now define the cost function $c(\rho, W)$ as

$$c(\rho, W) = Erl(\rho, W) \times \rho, \tag{4}$$

such that $c\left[\rho^{(k)}, W\right]$ represents the term in the numerator of the expression in Eq. (2) corresponding to link ℓ_k. Since the denominator in Eq. (2) is a constant, we can formulate our optimization problem in terms of a network flow model as follows:

$$\text{minimize } B_N \sum_{i \neq j} \rho_{ij} = \sum_{\ell_k \in E} c\left[\rho^{(k)}, W\right] \tag{5}$$

subject to:

$$\sum_{\ell_k \in tail(v)} x_{ij}^{(k)} - \sum_{\ell_k \in head(v)} x_{ij}^{(k)} = \begin{cases} 1, & \text{if } v = i \\ -1, & \text{if } v = j \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j, v, i \neq j, \tag{6}$$

$$\sum_{\ell_k \in tail(v)} x_{ij}^{(k)} \leq 1 \quad \forall i,j,v,\ i \neq j, \tag{7}$$

$$\sum_{\ell_k \in head(v)} x_{ij}^{(k)} \leq 1 \quad \forall i,j,v,\ i \neq j, \tag{8}$$

$$\rho^{(k)} = \sum_{ij} \rho_{ij}^{(k)} \times x_{ij}^{(k)} \quad \forall i,j,k\ i \neq j, \tag{9}$$

$$0 \leq x_{ij}^{(k)} \leq 1 \quad \forall i,j,k\ i \neq j. \tag{10}$$

The first three sets of constraints (6), (7), (8) represent the conservation of traffic flow at switch S_v. The fourth set of constraints (9) ensures that a traffic component contributes to the load of a link ℓ_k, if and only if some nonzero fraction $x_{ij}^{(k)}$ of this component travels over link ℓ_k.

In the formulation (5), (6), (7), (8), (9), (10), we have let the variables $x_{ij}^{(k)}$ be real numbers. Therefore, a solution to the problem might dictate that traffic between a given source–destination pair follows two or more different paths across the network. Of course, it is possible to restrict $x_{ij}^{(k)}$ to take only two possible values, 0 or 1; in this case, the solution will yield a single path for each traffic component. We note that restricting $x_{ij}^{(k)}$ to (binary) integer values could result in a worse solution (i.e., higher overall burst drop probability), and will also affect the computational complexity of the problem. We will revisit this issue in Subsection 3.B.

Clearly, the objective (5) is not a linear function of the variables $x_{ij}^{(k)}$. Therefore, as our last step toward a linear formulation, we will approximate the objective function by a piecewise linear function. Let us refer to Fig. 1 that plots the cost function $c(\rho, W)$ versus the value of ρ, when the number of wavelengths is $W = 32$. It is straightforward to show that, as seen in Fig. 1, the cost function is convex. The problem of fitting a convex curve by use of a piecewise linear function has been studied in Refs. [11–13], where the objective was to achieve the best least-squares fit. However, such an approach has two disadvantages if applied to the cost function $c(\rho, W)$: it is computationally expensive, and it could result in a large number of line segments that in turn would increase the complexity of solving the optimization problem.

Therefore, we decided to use simple interpolation to find a piecewise linear function to approximate the cost function $c(\rho, W)$. For example, let us assume that we use K line segments whose ρ coordinates fall within the range of $[0 = \rho_0, \rho_1), [\rho_1, \rho_2), \ldots, [\rho_{K-1}, \rho_K]$, where ρ_K is an upper bound on the load offered to any link in the network. Then the approximate linear cost function $\hat{c}(\rho, W)$ is given by

$$\hat{c}(\rho, W) = \frac{[c(\rho_m, W) - c(\rho_{m-1}, W)]}{\rho_m - \rho_{m-1}} (\rho - \rho_{m-1})\, \rho_{m-1} \leq \rho < \rho_m, \quad m = 1, 2, \ldots, K. \tag{11}$$

It should be clear that, if we use the above approximate cost function $\hat{c}(\rho, W)$ in place of $c(\rho, W)$ in the objective (5), the formulation (5), (6), (7), (8), (9) is a linear programming problem.

The number of line segments used in the approximate cost function $\hat{c}(\rho, W)$ represents a trade-off between the quality of approximation and the complexity of computation. Note that our objective is simply to reduce the load of links in the high-load and moderate-load regions, by increasing the load of links in the low-load region; because of the convex property, doing so will help reduce the overall cost (burst drop probability). Therefore, the approximate piecewise linear function should adequately capture the behavior of the cost function in the low-, moderate-, and high- load regions. From Fig. 1 we observe that, at low

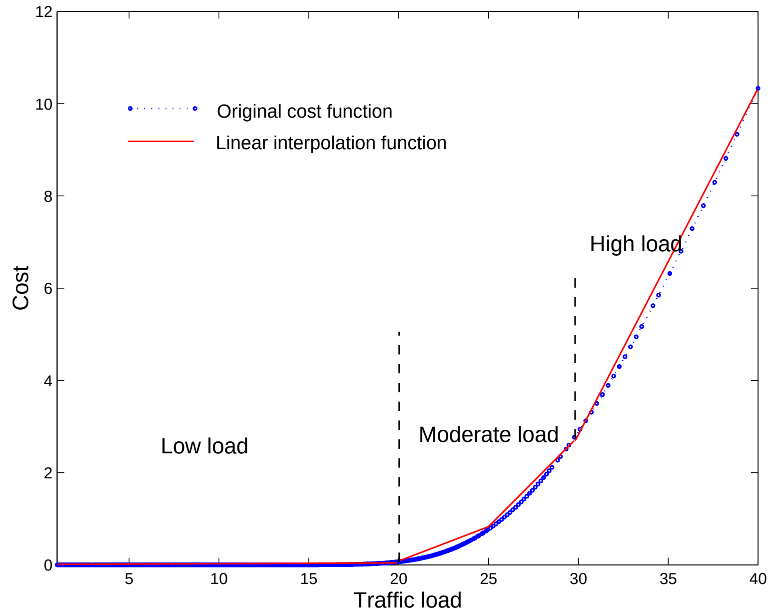

Fig. 1. Cost function $c(\rho, W)$ for $W = 32$.

and high loads, the cost function $c(\rho, W)$ resembles a straight line; similar observations can be made for other values of W (note that in Fig. 1 we used $W = 32$). Based on these observations, we used $K = 4$ segments in our approximation. When $W = 32$, we select the ρ coordinates for these four line sections as follows. The first line segment captures the behavior of the cost function at low loads, and its ρ coordinates are in the range $[0, \rho_1 = 20)$. The fourth line segment captures the cost behavior at high loads. To determine the ρ coordinates for this segment we let $\rho_4 = 40$ as a reasonable upper bound for the load on any network link, and we chose $\rho_3 = 30$ so that the slope of the cost function at $\rho = \rho_4$ is within 20% of the slope at $\rho = \rho_3$, i.e., $\vartheta c(\rho, W)/\vartheta\rho\,|_{\rho=\rho_4} \leq 1.2 \times \vartheta c(\rho, W)/\vartheta\rho\,|_{\rho=\rho_3}$; this constraint ensures that this part of the cost function can be accurately approximated by a straight line. Finally, we select $\rho_2 = 25$ as the midpoint between ρ_1 and ρ_3. A similar approach can be used to determine the line segments for other values of the number W of wavelengths.

Although we used a small number of segments $(K = 4)$ in the linear interpolation to approximate the cost function $c(\rho, W)$, it is certainly practical to use a larger number of segments to obtain a better approximation. For example, if after solving the problem with a small number of segments it is determined that the optimal value of the objective function in Eq. (5)is quite small, one might use a larger number of segments to better approximate the low-load region of the cost function. Returning to Fig. 1, one possible approach to determination of the number of segments K is the following. If after using $K = 4$ segments it is found that the traffic load along some link(s) is less than the value of ρ_1 used in approximating the corresponding cost function as in Fig. 1, one can increase the number of segments to $K = 5$ by splitting the segment $[0, \rho_1)$ into two segments $[0, \rho_1/2)$ and $[\rho_1/2, \rho_1)$. The problem can then be solved again, and the process repeated by splitting the first segment again, as necessary, to capture the effects of the low loads; a similar process can be used for high loads. However, we determined that the use of four segments yields satisfactory results.

We also note that the approach we described above to obtain a piecewise linear approximation of the objective function (5) can be used in the case of non-Poisson burst traffic models, as long as both the link and the overall burst drop probabilities can be expressed as a convex function of the link loads. Since we are not aware of any studies that have obtained analytical expressions for the overall burst drop probability, we consider only the objective function (5). We emphasize, however, that the optimized paths that we obtain by solving the formulation (5), (6), (7), (8), (9), (10) under the stated assumptions will benefit any OBS network regardless of the actual traffic arrivals.

3.B. Solving the Optimization Problem

The formulation (5), (6), (7), (8), (9), (10) contains $O\left(N^2M\right)$ variables $x_{ij}^{(k)}$ and $O\left(N^2M\right)$ constraints, where N is the number of switches (nodes) and M is the number of links in the OBS network. As long as we let variables $x_{ij}^{(k)}$ be real numbers as in Eq. (10) we obtain a linear programming (LP) problem that can be solved efficiently with the SIMPLEX algorithm [14] even for very large networks with hundreds of nodes and links. A solution to this LP model might dictate that traffic between some source–destination pairs takes multiple paths across the network. There are two issues that must be addressed with such a solution. First, consider a switch at which a given traffic component (i.e., the traffic between a given source–destination pair) must be split to take two or more different paths. The fraction of traffic that must be sent over each path must be equal to the corresponding traffic flow variables $x_{ij}^{(k)}$ obtained by solving the LP. Accomplishing this goal while taking into account other important constraints (e.g., preserving the order of packets contained in the bursts at the destination) can be a potentially challenging task. Second, the paths obtained through a solution to this optimization problem can be of a quite general form. In particular, it is possible that the various paths for a given source–destination pair split at some switch, merge at another downstream switch, split again later, etc. Such paths could pose challenges in configuring and maintaining consistency among the routing tables in the network.

To avoid the difficulties associated with multipath routing in OBS networks, we restrict all traffic between any given source–destination pair to be routed over a unique path. To this end we could modify the constraints (10) to restrict variables $x_{ij}^{(k)}$ to take only two values, 0 or 1. In this case, the solution will yield a single path for each source–destination pair. The formulation (5), (6), (7), (8), (9), (10) now becomes an integer linear programming (ILP) model that can be solved optimally by use of the CPLEX mathematical programming optimizer [15]. Once the optimal flow vector $\left\{x_{ij}^{(k)}\right\}$ has been determined with CPLEX, we can obtain the unique path for each source–destination (i,j) by using the algorithm in Algorithm 1. However, given the large number of variables and constraints, an optimal solution can be obtained only for networks of moderate size.

Algorithm 1 Algorithm for Computing Paths from the Traffic Flow Vector

---

Input: Flow vector $x_{ij} = \{x_{ij}^{(1)}, x_{ij}^{(2)}, \ldots, x_{ij}^{(M)}\}$ corresponding to traffic between switch S_i and S_j, $M = |E|$
Output: Path π_{ij} for all traffic between S_i and S_j

```
π_ij ← ∅
m ← i
while m ≠ j do
  for all ℓ_k ∈ tail(m) do
    if x_mn^(k) = 1 then
      Append ℓ_k to π_ij
      m ← n
```

For large networks with hundreds of nodes and links, it is not possible to obtain an optimal solution to the ILP within a reasonable amount of time. In this case, we propose the following simple greedy heuristic to obtain a unique path for each source–destination pair; we have found that this algorithm yields good results.

- Solve the corresponding LP by use of SIMPLEX to obtain a traffic flow vector with real values for $x_{ij}^{(k)}$.
- For each source–destination pair (i,j) with only one path under the LP solution, assign this path for routing bursts. Evaluate the objective function (5) by considering only paths that have been assigned a path so far.
- Sort the source–destination pairs not yet assigned a path according to the number of paths each has under the LP solution; break ties by sorting source–destination pairs in decreasing order of the length of their shortest path.
- Consider the first source–destination pair (i,j) not yet assigned a path; the traffic of this pair is split among n paths in the LP solution. For each of the n paths, evaluate the objective function (5) as if all traffic between i and j is sent over this path. Assign to pair (i,j) the path that minimizes the objective function; in other words, set the variables $x_{ij}^{(k)}$ along this path to 1, and set all other nonzero variables $x_{ij}^{(k)}$ in the LP solution to 0.
- Repeat the previous step until all source–destination pairs have been assigned a path.

4. Numerical Results

Here we use simulation to demonstrate the performance improvements that are possible when routing bursts along paths obtained through our optimization techniques, over the use of shortest paths. We use the simulator we developed as part of the JumpStart project [16]. The simulator accounts for all the details of the JumpStart OBS signal protocol [7], which employs the just-in-time (JIT) reservation scheme [17], including all messages required to set up the path of a burst and feedback messages from the network; the JumpStart signaling protocol has been implemented in a proof-of-concept test bed on the Advanced Technology Demonstration Network (ATDNet) [18]. [We emphasize, however, that the optimized routing paths we develop and evaluate in this work are independent of the specifics of the reservation protocol and can be deployed alongside either the just-enough-time (JET) or the Horizon reservation scheme.] 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 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 to improve readability.

In our simulations, we consider two different arrival processes for the generation of bursts. The first is a Poisson process, which is consistent with the assumptions we made in Section 3 to obtain the linear programming formulation. The second is the three-state Markovian process we developed and analyzed in Ref. [19], whose parameters can be selected to introduce any degree of burst into the arrival process. In our simulation, we assume that burst lengths are exponentially distributed; however, we have found that the actual burst length distribution does not have any significant effect on the results.

We compare three different fixed path routing schemes:

- **SP routing:** bursts are routed over the shortest path (in terms of hops) between source and destination, with ties broken arbitrarily.

- **LP routing:** solve the LP of Section 3 to obtain a set of paths for each source–destination pair; then use the heuristic in Subsection 3.B to assign a single path for routing bursts to each source–destination pair.

- **ILP routing:** bursts are routed over the paths determined by solving the ILP that corresponds to formulation (5), (6), (7), (8), (9), (10) after we modify the constraints (10) to restrict variables $x_{ij}^{(k)}$ to take only the value of 0 or 1; we were able to solve the ILP with CPLEX only for networks of moderate size. [A note on the running time requirements of the ILP and LP routing approaches is warranted. The LP model, along with the associated heuristic in Subection 3.B to determine a unique path for each source–destination pair, takes less than 1 s to run for the three networks we used in our simulations. The running time of the ILP model is between 30 and 90 min for the 16-node networks we consider in Section 4.A, depending on the network load (i.e., the ρ_{ij} values in the formulation (5), (6), (7), (8), (9), (10)). This running time is reduced by approximately 20% by termination of the search after a solution has been found to within 5% of the optimal value (refer also to Subsection 4.A). However, for the 33-node network we consider in Subsection 4.B, the ILP model did not terminate after running for two full days even with the 5% cutoff. All the running times were obtained on a SUN Ultra workstation with the Solaris 5.8 operating system. Therefore, LP routing is several orders of magnitude faster than ILP routing and can scale to networks of realistic size.]

We also 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 amount of traffic between a pair of switches is inversely proportional to the minimum number of hops between these two switches.

4.A. *Results for Networks of Moderate Size*

We first consider two 16-node networks: the 4×4 torus network shown in Fig. 2 is based on a regular topology, and the network in Fig. 3 is based on an irregular topology derived from the 14-node National Science Foundation (NSF) network. We emphasize that, even for these networks of moderate size, solving the ILP to obtain an optimal set of paths may take a long time (more than a few hours). Therefore, we utilize a feature of CPLEX that allows us to terminate the search once a solution that is within 5% of the optimal value has been found; this solution is close to the optimal and can be obtained in less time than required to obtain the optimal value. All the figures in this section plot the burst drop probability versus the normalized network load ρ_W, which is obtained by dividing the total load offered to the network by the number W of wavelengths: $\rho_W = \sum \rho_{ij}/W$.

4.A.1. Poisson Traffic

Figure 4 plots the burst drop probability versus ρ_W for the NSF network under the three routing schemes; these results were obtained with Poisson arrivals and the uniform traffic pattern. As can be seen, use of the paths obtained through our optimization approach (LP and ILP routing) outperforms the shortest path routing over the entire range of values for the normalized network load we considered. In the low-load region, the burst drop probability under optimized routing is as much as an order of magnitude lower than that under the shortest path routing. At moderate loads, the decrease in drop probability remains significant (as much as 50%); even at high loads, the use of paths to balance the load across network links can have a small benefit. Another important observation is that solving the

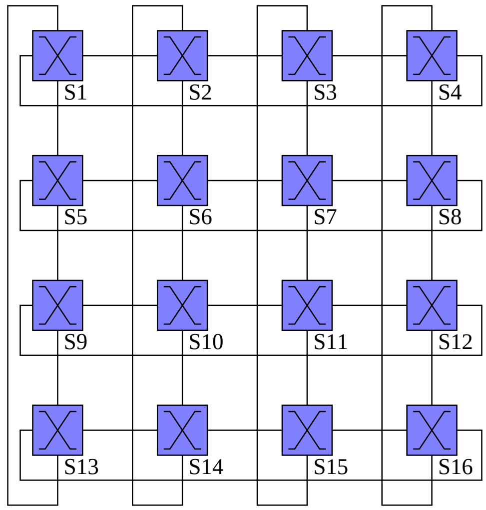

Fig. 2. The 4 × 4 torus network.

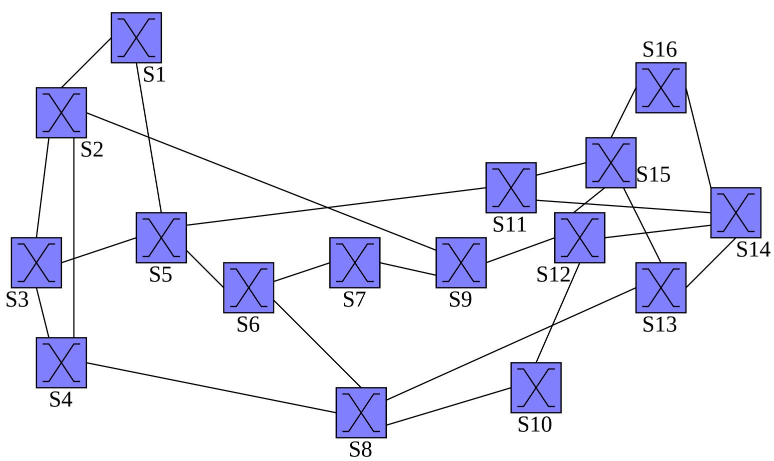

Fig. 3. Sixteen-node topology based on the 14-node NSF network.

ILP formulation to obtain the optimal paths has only a slight advantage over solving the LP formulation (which is orders of magnitude faster) and then use of the heuristic in Subsection 3.B to assign a single path to each source–destination pair. This result can be explained by the fact that, in this case, the solution to the LP formulation yields one or two paths for each source–destination pair; and for most pairs with two paths, one of the paths is dominant, carrying the vast majority of the traffic.

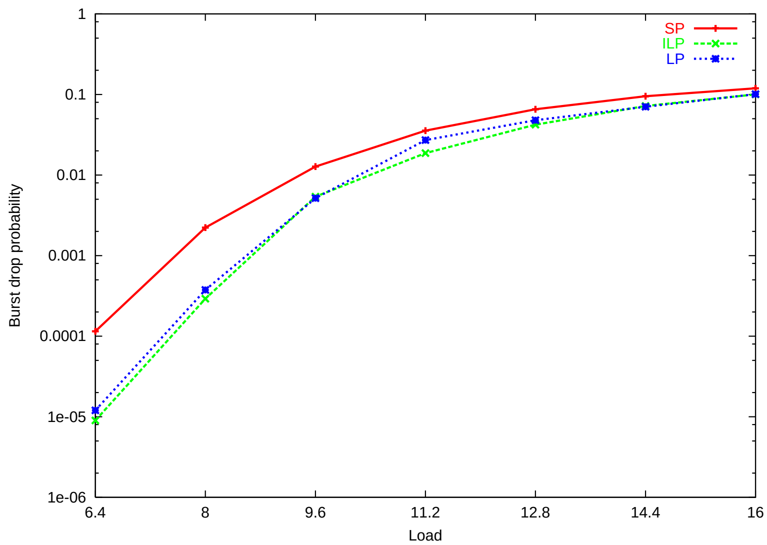

Fig. 4. Burst drop probability NSF network, Poisson arrivals, uniform traffic pattern.

Figure 5 presents simulation results of the NSF network with the distance-dependent traffic pattern. There are two important observations we can make from this figure. First, we note that the solution obtained by use of the LP followed by the simple path assignment heuristic in Subsection 3.B outperforms the one obtained by the ILP. This result is due to the fact that, in this case, as we explained earlier in this section, we were unable to obtain an optimal solution with CPLEX but rather a suboptimal one. The second observation is that, under the same normalized network load, the improvement in burst drop probability over shortest path routing is significantly higher for the distance-dependent traffic pattern of Fig. 5 compared with the uniform pattern of Fig. 4. Since shortest-path routing uses the same set of paths regardless of the actual traffic pattern, its performance under the same network load is similar under either pattern. However, our optimization approach uses the information about the traffic pattern to tailor the routing paths in a way that appropriately balances the load across the network links. As a result, for the distance-dependent pattern in Fig. 5, the burst drop probability is reduced by up to 2 orders of magnitude under low and moderate loads and almost 1 order of magnitude under high loads.

Figures 6, 7 show the results for the torus network for uniform and distance-dependent traffic, respectively. We again find that optimized routing performs significantly better than shortest path routing. For the reasons we explained above, this improvement in performance is higher under the distance-dependent traffic pattern. We also observe that LP routing closely tracks ILP routing (or slightly outperforms it when the ILP is solved suboptimally), similar to the behavior we observed with the NSF network. Comparing the two figures to the corresponding figures for the NSF network, we note that, under the same size network load, the burst drop probability is lower in the Torus network compared with the NSF network because of the symmetry of the torus topology. Because of the topology's inherent load balancing properties, even the shortest path routing performs well compared with asymmetric topologies such as the NSF network. However, we also observed that, even with such a symmetric topology, our optimization approach can further exploit the information with regard to the traffic pattern to offer significant advantages over the shortest path routing.

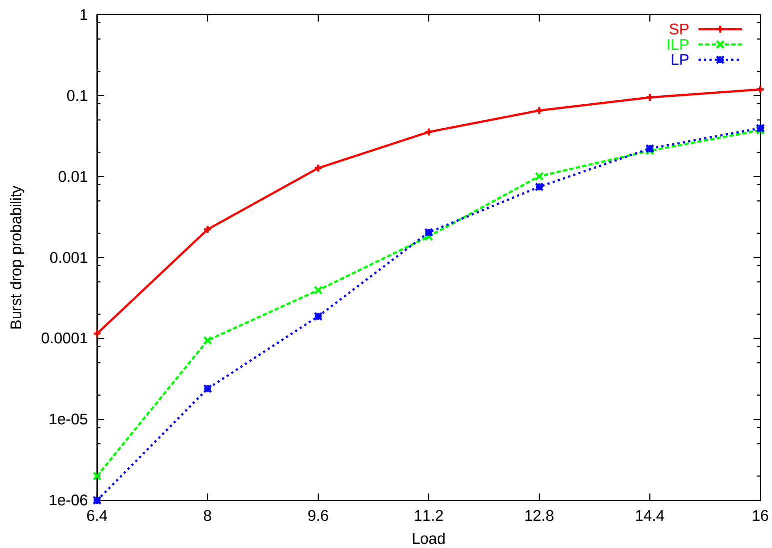

Fig. 5. Burst drop probability, NSF network, Poisson arrivals, distance-dependent traffic pattern.

4.A.2. Non-Poisson Traffic

In all the simulation results presented so far, burst traffic between each source–destination pair was generated according to a Poisson process with a parameter determined by the specific traffic pattern used. The Poisson arrival assumption is consistent with the approximations that led us to the linear problem formulation in Section 3. In this section we present simulation results for which we used a different arrival process to generate bursts. The arrival process we used is the three-state Markov process introduced in Ref. [19]. The process can be in one of three states: in the short burst (respectively, long burst) state, the user is in the process of transmitting a short (respectively, long) burst; in the idle state the user does not transmit any burst. It was shown in Ref. [19] that, by appropriate selection of the parameters of the process (i.e., the mean duration of each state and the transition rates between states), it is possible to introduce any degree of burst into the arrival process. For the results we present here, the arrival process we used has significantly more burst than the Poisson process (the coefficient of variation is 3.5).

Figures 8, 9 are similar to Figs. 5, 7, respectively, except that the burst arrivals were generated by use of the three-state Markovian process [19] rather than a Poisson process. We used the distance-dependent traffic pattern to obtain these results; similar results were observed for the uniform pattern. The relative behavior of the three curves in Figs. 8, 9 is similar to that under Poisson traffic in that LP and ILP routing perform significantly better than SP routing. As can be seen, selection of the paths by use of the optimization techniques we developed under the Poisson process assumption produces significant benefits in terms of burst drop probability even when the arrival process is not Poisson.

Table 1 provides additional insight into the optimization approach we proposed and its ability to improve the burst drop probability over shortest path routing regardless of the

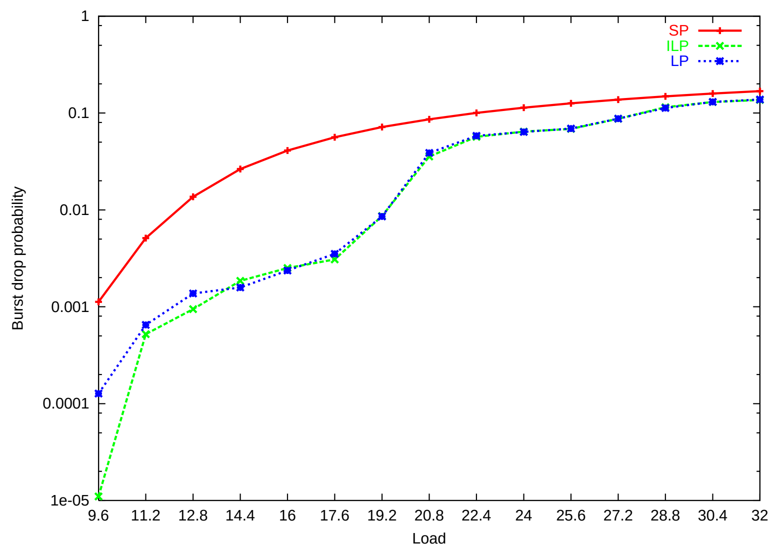

Fig. 6. Burst drop probability, Torus network, Poisson arrivals, uniform traffic pattern.

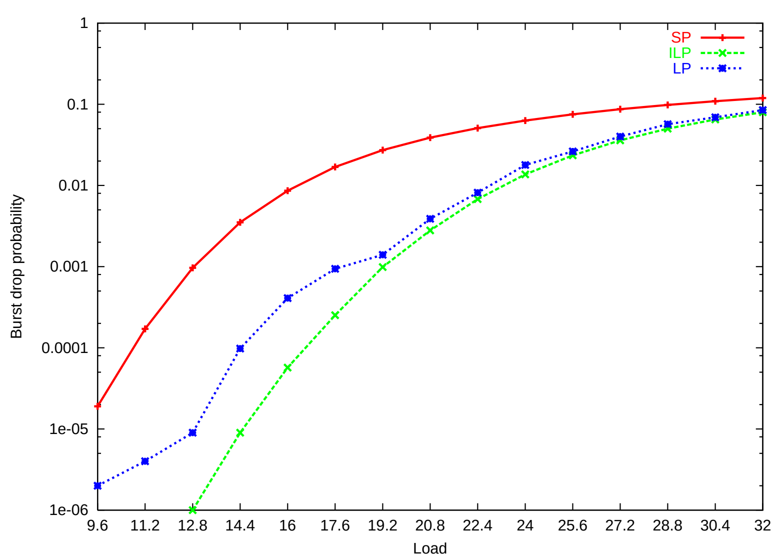

Fig. 7. Burst drop probability, Torus network, Poisson traffic, distance-dependent traffic pattern.

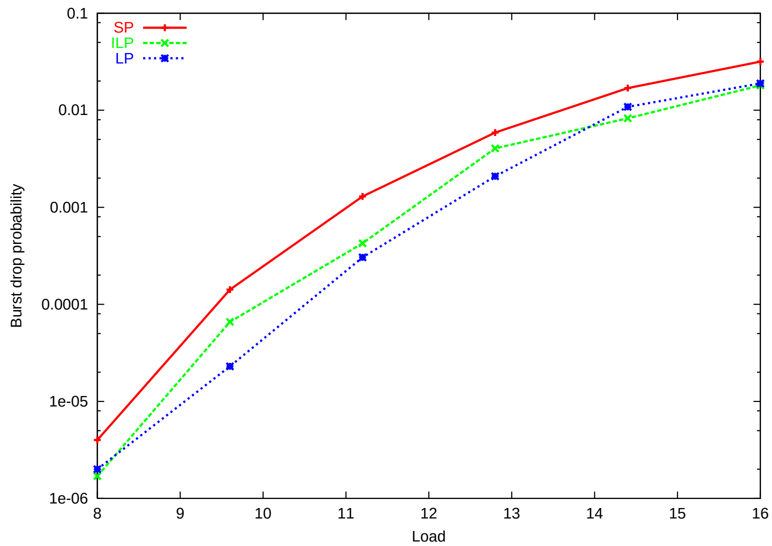

Fig. 8. Burst drop probability, NSF network, non-Poisson arrivals, distance-dependent traffic pattern.

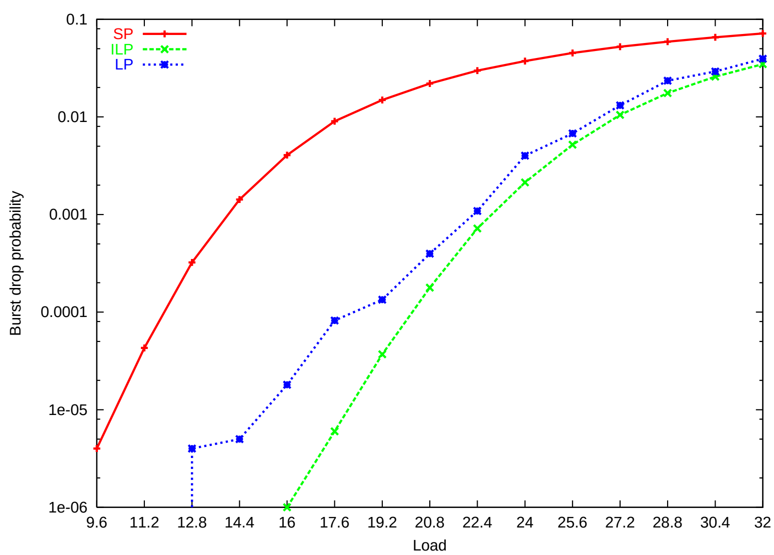

Fig. 9. Burst drop probability, Torus network, non-Poisson arrivals, distance-dependent traffic pattern.

topology, traffic pattern, or traffic arrival assumptions. The table lists the maximum and minimum link loading of the NSF and Torus networks under SP, LP, and ILP routing, and the stated traffic pattern and arrival process; the link load values for the NSF (respectively, Torus) network correspond to a normalized network load of 6.4 (respectively, 9.6). Note that the essence of the path optimization approach of Section 3 is to shift the traffic onto paths so as to reduce the load at the most utilized links while increasing the load of the least utilized ones. Although we used the Erlangian blocking model to simplify the formulation, the net effect of our approach is similar regardless of the traffic assumption. This is evident in Table 1, where we observe that LP and ILP routing results in a smaller maximum link load than SP routing. In general, a lower link loading will lead to a lower burst drop probability, regardless of the burst arrival process, as we have observed in this subsection.

Table 1. Minimum and Maximum Link Load under Each Routing Scheme

Routing Scheme		NSF Network		Torus Network	
		Uniform	Distance Dependent	Uniform	Distance Dependent
SP	Min	1.7	2.43	2.56	2.24
	Max	17.07	12.96	21.76	16.05
LP	Min	3.4	3.77	3.84	4.48
	Max	14.5	11.87	17.92	12.7
ILP	Min	0.85	2.96	7.68	7.28
	Max	14.5	10.8	14.08	8.96

4.B. Results for Large Networks

We now demonstrate the benefits of our path optimization approach by considering a large network topology for which it is not possible to solve the ILP formulation to obtain the optimal fixed paths. Therefore, here we compare shortest path routing to routing over paths obtained by solving the LP formulation and then rounding the traffic flow variables as we explained in Section 3.B. In our simulations, we used the 33-node topology shown in Fig. 10. This topology is based on the 33-node multigigabit pan-European research network as of April 2004 (see http://www.geant.net), but we added the links represented by dashed lines in Fig. 10 to ensure that the network is biconnected.

Figures 11, 12 plot the burst drop probability of SP and LP routing for the Gigabit European Academic Network (GEANT) and the distance-dependent traffic pattern; Figure 11 shows the results when the arrival process is Poisson, whereas the results of Fig. 12 were obtained by generating bursts according to the three-state Markov process we discussed earlier. As can be seen, LP routing outperforms SP routing by a wide margin except at high loads. Furthermore, this observation is true regardless of the arrival process (Poisson or not). Since the LP routing optimization procedure is quite fast even for large networks, we conclude that our techniques can be applied in a practical and efficient manner to improve the burst drop probability in networks of any size.

5. Concluding Remarks

We have addressed the problem of selecting paths in an optical burst switching network to minimize the overall burst drop probability. We have taken a traffic engineering approach for which the objective has been to balance the burst traffic as much as possible across the network links. We developed an approximate formulation as an integer linear optimization problem by making some simplified assumptions. We have also presented a heuristic that

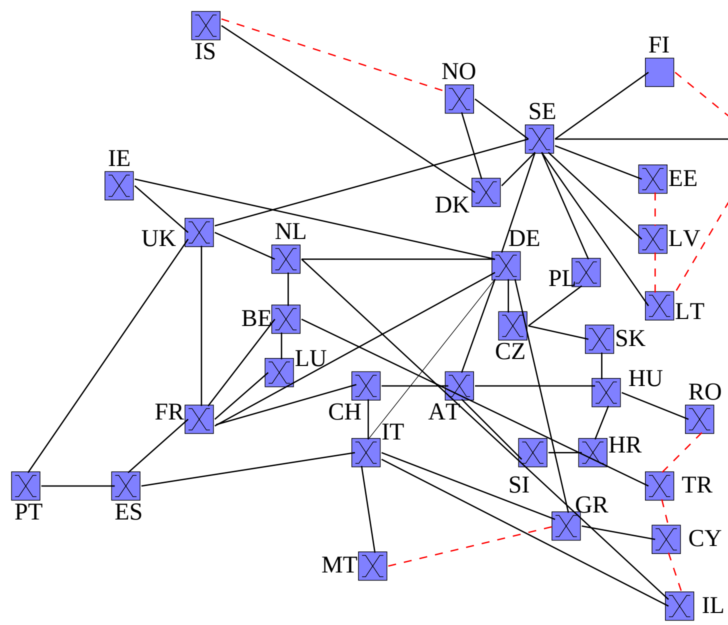

Fig. 10. The 33-node topology based on the 33-node GEANT network.

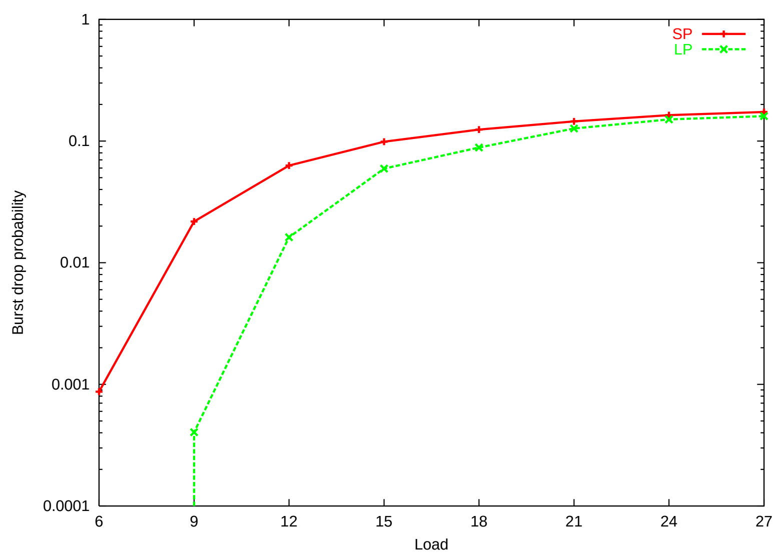

Fig. 11. Burst drop probability, GEANT network, Poisson traffic, distance-dependent traffic pattern.

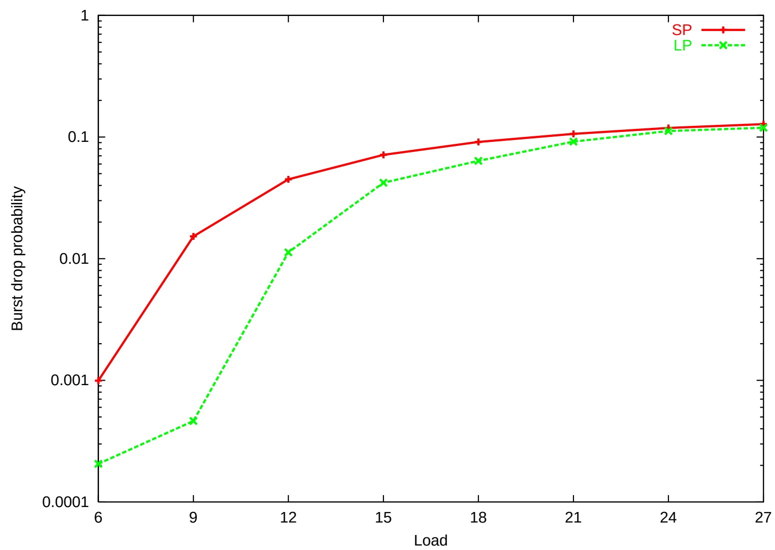

Fig. 12. Burst drop probability, GEANT network, non-Poisson traffic, distance-dependent traffic pattern.

allows us to solve the problem efficiently, albeit suboptimally, for large networks. Our results indicate that our approach is successful in obtaining paths that balance the load evenly, leading to a reduction in the burst drop probability for networks of various sizes and topologies, different traffic patterns, and burst arrival processes.

Acknowledgments

This material is based on research supported by the MCNC Research and Development Institute as part of the JumpStart project. This is an extended version of a paper that appeared in the Proceedings of Optical Network Design and Modeling 2005.

This study was performed while J. Teng was with the Department of Computer Science, North Carolina State University.

References and Links

[1] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: a new area in optical networking research," IEEE Network **18**(3), 16–23 (2004).
[2] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, "A performance model of deflection routing in multibuffer networks with nonuniform traffic," IEEE/ACM Trans. Netw. **3**(5), 509–520 (1995).
[3] X. Wang, H. Morikawa, and T. Aoyama, "Burst optical deflection routing protocol for wavelength routing WDM networks," in *Proceedings of Opticomm* (2003), pp. 120–129.
[4] C. Hsu, T. Liu, and N. Huang, "Performance analysis of deflection routing in optical burst-switched networks," in *Proceedings of Infocom*, pp. 66–73 (2002).
[5] 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.
[6] J. Iness, B. Ramamurthy, B. Mukherjee, and K. Bala, "Elimination of all-optical cycles in wavelength-routed optical networks," J. Lightwave Technol. **14**(6), 1207–1217 (1996).
[7] 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. **40**(2), 82–89 (2002).

[8] R. Dutta and G. N. Rouskas, “A survey of virtual topology design algorithms for wavelength routed optical networks,” Opt. Netw. **1**(1), 73–89 (2000).
[9] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications* (Prentice-Hall, 1993).
[10] Z. Rosberg, H. L. Vu, M. Zukerman, and J. White,” “Performance analyses of optical burst-switching networks,” IEEE J. Sel. Areas Commun. **21,** 1187–1197 (2003).
[11] R. Bellman, “On the approximation of curves by line segments using dynamic programming,” Commun. ACM **4**(6), 284 (1961).
[12] B. Gluss, “Further remarks on line segment curving-fitting using dynamic programming,” Commun. ACM **5**(8), 441–443 (1962).
[13] H. Stone, “Approximation of curves by line segments,” Math. Comput. **15**(1), 40–47 (1961).
[14] R. J. Vanderbei, *Linear Programming: Foundations and Extensions* (Springer, 2001).
[15] http://www.ilog.com//products/cplex/.
[16] J. Teng, “A study of optical burst switched networks with the JumpStart just-in-time signaling protocol,” Ph.D. dissertation (North Carolina State University, Raleigh, N.C., 2004).
[17] J. Y. Wei and R. I. McFarland, Jr., “Just-in-time signaling for WDM optical burst switching networks,” J. Lightwave Technol. **18,** 2019–2037 (2000).
[18] 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*, Vol. 5, pp. 2777–2781, December 2003.
[19] 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*, pp. 2019–2029 (April 2003).