Routing Path Optimization in Optical Burst Switched Networks

Jing Teng and George N. Rouskas
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
E-mail: {jteng,rouskas}@eos.ncsu.edu

***Abstract*— It is usually assumed that optical burst switched (OBS) networks employ shortest path routing along with next-hop burst forwarding. Shortest path routing minimizes delay and optimizes the utilization of resources, however, it often causes certain links to become congested while others remain underutilized. In a buffer-less OBS network in which burst drop probability is the primary metric of interest, the existence of a few highly congested links may lead to unacceptable performance for the entire network. In this paper, we take a traffic engineering approach to path selection in OBS networks with the objective of balancing the traffic across the network links in order to reduce congestion and 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 shortest path routing.**

I. Introduction

Optical burst switching (OBS) is a promising switching paradigm which aspires to provide a flexible infrastructure for carrying future Internet traffic in an effective yet practical manner. OBS separates the control (signaling) and data plane functions in the network in a way that exploits the distinct advantages of optical and electronic technologies. Signaling messages are processed electronically at every node in the network, while bursts are transmitted transparently end-to-end, without 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 consume a full wavelength. The transmission of each burst is preceded by the transmission of a setup message (also referred to as burst-header control message), whose purpose is to reserve switching resources along the path for the upcoming data burst. An OBS source node does not wait for confirmation that an end-to-end connection has been set-up; instead it starts transmitting a data burst after a delay (referred to as "offset"), following the transmission of the setup message.

Over the last 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 [6]. 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 investigating 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 scenario is highly undesirable in OBS networks in which burst drop probability is the primary performance metric of interest: since it is generally assumed that intermediate switches do not buffer bursts, having a few highly congested links may lead to unacceptably high burst loss for the entire network.

One possible routing mechanism that can be used to reduce the burst loss due to sub-optimal path selection (e.g., shortest paths) is deflection routing [4]. 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 [13], while [9], [16] analyzed the performance of deflection routing. However, deflection routing in OBS networks has

0-7803-8956-5/05/$20.00 ©2005 IEEE.

several disadvantages. A practical implementation would require intermediate switches which deflect a burst to somehow increase its offset, an operation that is impossible without the use of buffers (alternatively, each burst must have an offset large enough to account for all possible deflections in its path, severely degrading the performance of the network). When deflection decisions are made at each switch without coordination with the rest of the network (a typical approach given the limited amount of time between the setup message and the data burst), there is great potential for routing loops which can have disastrous effects in an optical network [10]. Finally, deflection routing is by nature suboptimal since it only considers the congestion of the current switch, not the state of the links further along the path; and it may cause undesirable vibration effects, as explained in [16].

In this work, 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 in order to reduce congestion and 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 (which 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 to shortest path routing.

The rest of the paper is organized as follows. In Section II we discuss our assumptions regarding the OBS network we consider in our study. In Section III 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 IV, we present simulation results to demonstrate the effectiveness of our approach, and we conclude the paper in Section V.

II. The OBS 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, which generate optical bursts. An optical switch consists of two components: an optical cross-connect (OXC) which can optically forward a burst from an input to an output port without OEO conversion; and a signaling engine which processes signaling messages and controls the OXC switching fabric. Optical fiber links interconnect the network of switches, and also connect each user to one or more edge switches. A burst generated by a user travels past a series of fibers and switches in the OBS network, and terminates at another user.

We will 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 $tail(i) = \{\ell_k | S_i \text{ is the tail of } \ell_k\}$, as the set of links with S_i as their tail; similarly $head(i) = \{\ell_k | S_i \text{ is the head of } \ell_k\}$ 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, $\{\lambda_1, \lambda_2, \cdots, \lambda_W\}$.

We assume that the OBS network employs source routing, in that the ingress switch (source) determines the path of a burst entering the network. The path over which the burst must travel is carried by the setup message that precedes the transmission of the data burst. The network uses either fixed-path or multi-path routing. In fixed-path routing, all bursts between a source-destination pair follow the same path through the network. In multi-path routing, a burst may 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 which are used in the case of wavelength contention. The network does not use any other contention resolution mechanism. Specifically, OBS switches do not employ any buffering, either electronic or optical, in the data path, and they do not utilize deflection routing. 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.

III. Path Optimization for OBS Networks

We take a traffic engineering approach to computing 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 a $N \times N$ matrix $\Gamma = [\gamma_{ij}]$, where γ_{ij} represents the (long-term) arrival rate of bursts originating at switch S_i and destined for switch S_j. The values of the traffic elements γ_{ij} can be obtained empirically, or they can be based on predictions regarding the long-term demands placed upon the network; while these values may 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 traveling 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 determining a set of paths that optimize a certain performance metric of interest (e.g., congestion, average delay, etc.) is to formulate and solve an optimization problem (refer to [7] and references thereof for similar problems in wavelength routed networks). We take a similar approach in that we formulate a linear optimization problem in order 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 $\{\rho_{ij}, i, j = 1, \cdots, N\}$. However, we note that the problem at hand is different than typical network flow problems [1] 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;
- 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, which are *unknown* and can be determined only once the optimal paths have been obtained (the actual load on a link 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); and
- 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, self-similar, etc.); for non-Poisson burst arrival models, this relationship may be difficult (or even impossible) to express analytically.

Next, we present a formulation which overcomes the above difficulties and allows us to obtain routing paths which improve the burst drop probability significantly over 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 in order to obtain a linear model and avoid complex and computationally expensive formulations. The following discussion explains our assumptions and notations.

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:

$$b(\pi) \;=\; 1 - \prod_{\ell_k \in \pi} (1 - \beta^{(k)}) \;\simeq\; \sum_{\ell_k \in \pi} \beta^{(k)} \;\ll\; 1 \quad (1)$$

Therefore, we will assume that the actual traffic load $\hat{\rho}_{ij}$ seen by the network due to traffic originating at switch S_i and terminating 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, which is more accurate when the burst drop probability is low, but one which 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 burst traffic in the network is given by:

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

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 expression (2). As we mentioned previously, however, the expression for B_N depends on the burst arrival model. In general, it may not be possible to express B_N as a linear function of $x_{ij}^{(k)}$, and in fact, it may 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 using our approach will help reduce the burst drop probability (compared to schemes such as shortest-path routing) since they tend to more evenly balance the load 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:

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

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

$$c(\rho, W) = Erl(\rho, W) \times \rho \quad (4)$$

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

$$\textbf{minimize} \quad B_N \sum_{i \neq j} \rho_{ij} = \sum_{\ell_k \in E} c(\rho^{(k)}, W) \quad (5)$$

subject to:

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

(7)

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

$$\sum_{\ell_k \in head(v)} x_{ij}^{(k)} \leq 1 \quad \forall i, j, v \;\; i \neq j \quad (9)$$

$$\rho^{(k)} = \sum_{ij} \rho_{ij}^{(k)} \times x_{ij}^{(k)} \quad \forall\, i, j, k \;\; i \neq j \quad (10)$$

$$0 \leq x_{ij}^{(k)} \leq 1 \quad \forall i, j, k \;\; i \neq j \quad (11)$$

The first three sets of constraints (7)-(9) represent the conservation of traffic flow at switch S_v. The fourth set of constraints (10) ensures that a traffic component contributes to the load of a link ℓ_k if and only if some non-zero fraction $x_{ij}^{(k)}$ of this component travels over link ℓ_k.

In the above formulation, 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 follow 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 may 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 the next subsection.

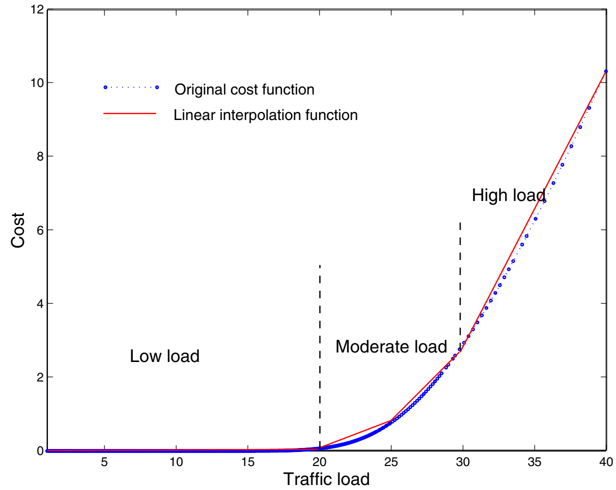

Fig. 1. Cost function $c(\rho, W)$ for $W = 32$

Clearly, the objective (5) is not a linear function of the variables $x_{ij}^{(k)}$. Therefore, as our last step towards a linear formulation, we will approximate the objective function by a piecewise linear function. Let us refer to Figure 1 which plots the cost function $c(\rho, W)$ against the value of ρ, when the number of wavelengths is fixed to $W = 32$. It is straight-forward to show that, as seen in the figure, the cost function is convex. The problem of fitting a convex curve using a piecewise linear function has been studied in [5], [8], [11], 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 may result in a large number of line segments, which in turn increases the complexity of solving the optimization problem.

Therefore, we have decided to use simple interpolation to find a piecewise linear function to approximate the cost function $c(\rho, W)$. For instance, let us assume that we use K line segments whose ρ coordinates fall in the range $[0 = \rho_0, \rho_1)$, $[\rho_1, \rho_2), \cdots, [\rho_{K-1}, \rho_K]$, respectively, 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:

$$\hat{c}(\rho, W) = \frac{(c(\rho_m, W) - c(\rho_{m-1}, W))}{\rho_m - \rho_{m-1}} (\rho - \rho_{m-1})$$
$$\rho_{m-1} \leq \rho < \rho_m, \quad m = 1, 2, \cdots, K \quad (12)$$

It should be clear that, if we use the above approximate cost function $\hat{c}(\rho, W)$ in the place of $c(\rho, W)$ in the objective (5), the formulation (5)-(10) is a linear programming problem.

The number of line segments used in the approximate cost function $\hat{c}(\rho, W)$ represents a tradeoff between the quality of approximation and the complexity of computation. Recall 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; due to 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 Figure 1, we observe that, at low and high loads, the cost function $c(\rho, W)$ resembles a straight line; similar observations can be made for other values of W (recall that in Figure 1 we have used $W = 32$). Based on these observations, we have used $K = 4$ segments in our approximation. In the case where $W = 32$, 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 choose $\rho_3 = 30$ so that the slope of the cost function at $\rho = \rho_4$ is within 20% of the slope at $\rho = \rho_3$, i.e., $\frac{\vartheta c(\rho, W)}{\vartheta \rho} |_{\rho=\rho_4} \leq 1.2 \times \frac{\vartheta c(\rho, W)}{\vartheta \rho} |_{\rho=\rho_3}$; this constraint ensures that this part of the cost function can be accurately approximated by a straight line. Finally, we select $\rho_2 = 20$ 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 have 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 instance, if, after solving the problem with a small number of segments it is determined that the optimal value of the objective function in (5) is quite small, then one might use a larger number of segments to better approximate the low-load region of the cost function. However, we have found that using four segments yields satisfactory results.

Finally, we 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 will only consider the objective function (5). We emphasize, however, that the optimized paths we obtain by solving the formulation (5)-(11) under the stated assumptions, will benefit any OBS network regardless of the actual traffic arrivals.

B. Solving the Optimization Problem

The formulation (5)-(11) contains $O(N^2M)$ variables $x_{ij}^{(k)}$ and $O(N^2M)$ 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 (11), we obtain a linear programming (LP) problem which can be solved efficiently using SIMPLEX 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 take 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 may pose challenges in configuring and maintaining consistency among the routing tables in the network.

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
```

Fig. 2. Algorithm for computing paths from the traffic flow vector

Alternatively, we could modify the constraints (11) 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. However, the formulation (5)-(11) now becomes an integer linear programming (ILP) model, and given the large number of variables and constraints, it can be solved optimally using CPLEX only for networks of moderate size. Once the optimal flow vector $\{x_{ij}^{(k)}\}$ has been obtained, we can obtain the path for each source-destination (i, j) using the algorithm in Figure 2.

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 which we have found to yield good results.

- Solve the corresponding LP using SIMPLEX to obtain a traffic flow vector with real values for $x_{ij}^{(k)}$.

- For each source-destination pair (i, j) that has 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 non-zero variables $x_{ij}^{(k)}$ in the LP solution to 0.
- Repeat the previous step until all source-destination pairs have been assigned a path.

IV. Numerical Results

In this section, we use simulation to demonstrate the performance improvements that are possible when routing bursts along paths obtained through our optimization techniques, over using shortest paths. We use the simulator we developed as part of the Jumpstart project [12]. The simulator accounts for all the details of the Jumpstart OBS signaling protocol [3] which employs the Just-In-Time (JIT) reservation scheme [14], including all messages required for setting up the path of a burst and feedback messages from the network; the Jumpstart signaling protocol has been implemented in a proof-of-concept testbed on the ATDNet [2]. (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 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; however, they are so narrow that we omit them from the figures we present in this section in order to improve readability.

In our simulations, we consider two different arrival processes for generating bursts. The first is a Poisson process, which is consistent with the assumptions we made in Section III to obtain the linear programming formulation. The second is the three-state Markovian process we developed and analyzed in [15], whose parameters can be selected to introduce any degree of burstiness 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 III to obtain a set of paths for each source-destination pair; then use the heuristic in Section III-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 corresponding to formulation (5)-(11) after we modify the constraints (11) to restrict variables $x_{ij}^{(k)}$ to take only the values 0 or 1; we were able to solve the ILP using CPLEX only for networks of moderate 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.

A. Results for Networks of Moderate Size

We first consider two 16-node networks: the 4×4 torus network shown in Figure 3 is based on a regular topology, while the network in Figure 4 is based on an irregular topology derived from the 14-node 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 terminate the search once CPLEX has found a solution that is within 5% of the optimal. All the figures in this section plot the burst drop probability against the "normalized network load" ρ_W, which is obtained by dividing the total load offered to the network by the number W of wavelengths: $\rho_W = \frac{\sum_{ij} \rho_{ij}}{W}$.

1) Poisson Traffic: Figure 5 plots the burst drop probability against ρ_W for the NSF network under the three routing schemes; these results were obtained with Poisson arrivals and the uniform traffic pattern. As we can see, using the paths obtained through our optimization approach (LP and ILP routing) outperforms 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 up to an order of magnitude lower than that under shortest path routing, while at moderate loads, the decrease in drop probability remains significant (up to 50%); even at high loads, using paths so

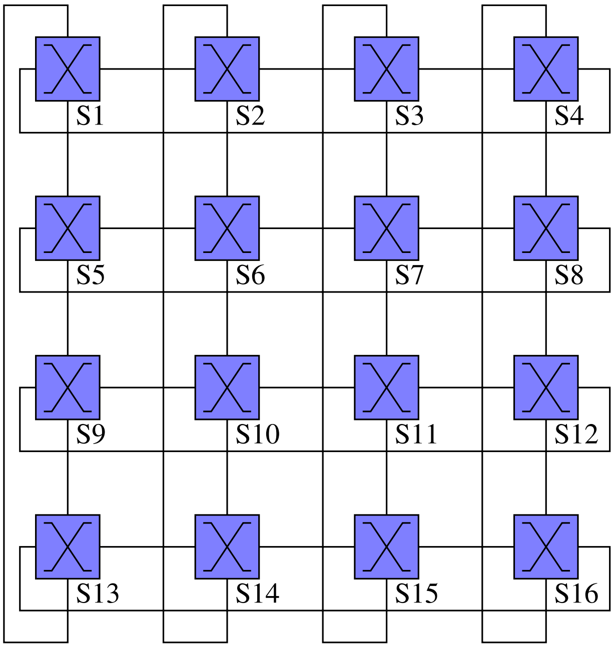

Fig. 3. The 4 × 4 torus network

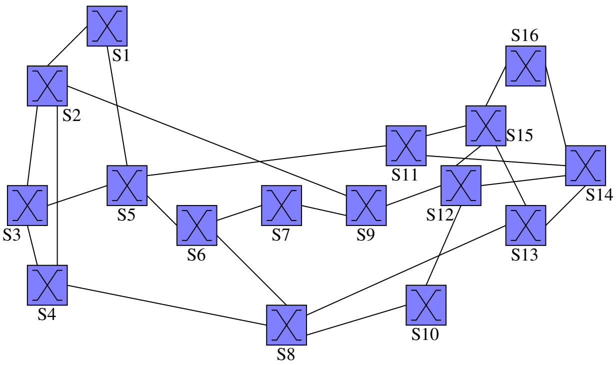

Fig. 4. The 16-node topology based on the 14-node NSFNet

as to balance the load across the network links can have a small benefit. Another important observation is that solving the ILP formulation to obtain the optimal paths has only a small advantage over solving the LP formulation (which is orders of magnitude faster) and then using the heuristic in Section III-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.

Figure 6 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 using the LP followed by the simple path assignment heuristic in Section III-B outperforms the one obtained by the ILP. This result is due to the fact that, in this case, as we explained earlier, we were not able 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 Figure 6 compared to the uniform pattern of Figure 5. 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 Figure 6, the burst drop probability is reduced by up to two orders of magnitude under low and moderate loads, and almost one order of magnitude under high loads.

Figures 7 and 8 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 sub-optimally), 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 normalized network load, the burst drop probability is lower in the Torus network compared to the NSF network. This is due to the symmetry of the torus topology; due to the topology's inherent load balancing properties, even shortest path routing performs well compared to asymmetric topologies such as the NSF network. However, we also see that, even with such a symmetric topology, our optimization approach can further exploit the information regarding the traffic pattern to offer significant advantages over shortest path routing.

2) Non-Poisson Traffic: In all the simulation results we have 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 III. In this section we present simulation results in which we have used a different arrival process to generate bursts. The arrival process we used is the 3-state Markov process introduced in [15]. The process may 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, while in the "idle" state the user is not transmitting any burst. It was shown in [15] that, by appropriately selecting the parameters of the process

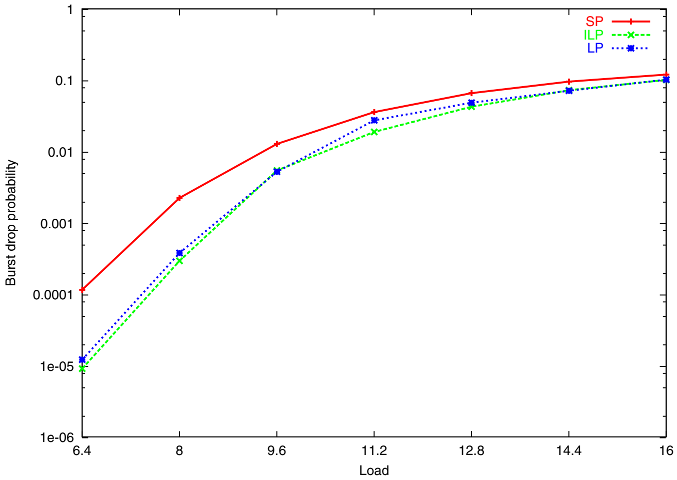

Fig. 5. Burst drop probability NSF network, Poisson arrivals, uniform traffic pattern

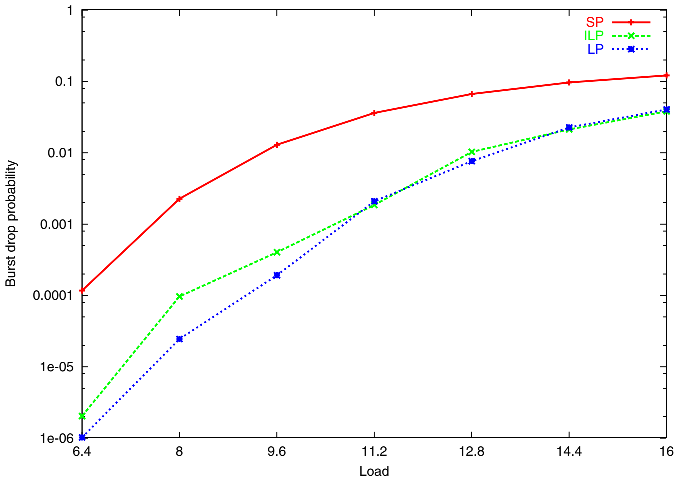

Fig. 6. Burst drop probability, NSF network, Poisson arrivals, distance-dependent traffic pattern

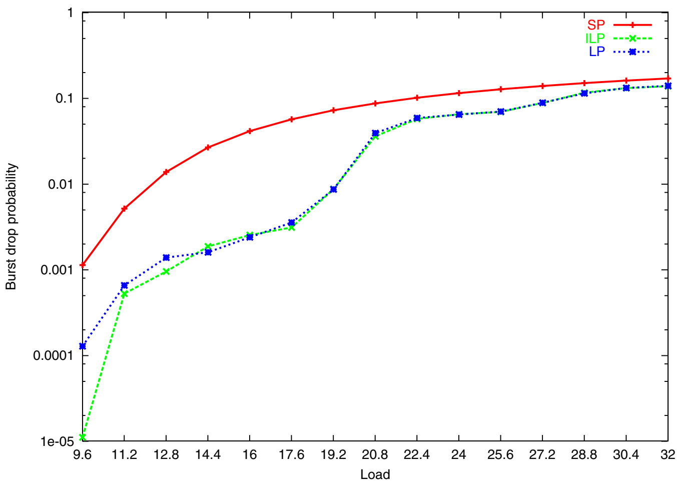

Fig. 7. Burst drop probability, Torus network, Poisson arrivals, uniform traffic pattern

(i.e., the mean duration of each state and the transition rates between states), it is possible to introduce any degree of burstiness into the arrival process. For the results we present in this section, the arrival process we used is significantly more bursty than the Poisson process (the coefficient of variation is 3.5).

Figures 9 and 10 are similar to Figures 6 and 8, respectively, except that in this case, burst arrivals were generated using the 3-state Markovian process of [15] rather than a Poisson process. The traffic pattern we used for obtaining these results is the distance-dependent one; very similar results were observed for the uniform pattern as well. The relative behavior of the three curves in Figures 9 and 10 is similar to that under Poisson traffic in that LP and ILP routing perform significantly better than SP routing. As we can see, selecting the paths using 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 I provides additional insight into the optimization approach we have proposed and its ability to improve the burst drop probability over shortest path routing regardless of the 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 NSF (respectively, Torus) network correspond to a normalized network load of 6.4 (respectively, 9.6). Recall that the essence of the path optimization approach of Section III is to shift the traffic onto paths so as to reduce the load at the most utilized links while increasing the load of 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 the above table, where we see that LP and ILP routing result in a smaller maximum link load than SP routing. In general, a lower link loading will lead to lower burst drop probability, regardless of the burst arrival process, as we have observed in this section.

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, in this section 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 III-B. In our simulations, we used the 33-node topology shown in Figure 11. This topology is based on the 33-node multi-gigabit pan-European research network as of April 2004 (see http://www.geant.net), but we added the links shown

Routing Scheme		NSF Network Uniform	NSF Network Distance-Dependent	Torus Network Uniform	Torus Network 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

TABLE I

MINIMUM AND MAXIMUM LINK LOAD UNDER EACH ROUTING SCHEME

in dashed lines in Figure 11 to ensure that the network is biconnected.

Figures 12 and 13 plot the bust drop probability of SP and LP routing for the GEANT network and the distance-dependent traffic pattern; Figure 12 shows the results when the arrival process is Poisson, whereas the results of Figure 13 were obtained by generating bursts according to the 3-state Markov process we discussed earlier. As we can see, LP routing outperforms SP routing by a wide margin except at very 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.

V. CONCLUDING REMARKS

We have addressed the problem of selecting paths in an OBS network in order to minimize the overall burst drop probability. We have taken a traffic engineering approach where the objective has been to balance the burst traffic as much as possible across the network links. We have developed an approximate formulation as an integer linear optimization problem by making some simplified assumptions. We have also presented a heuristic that allows us to solve the problem efficiently, albeit sub-optimally, 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.

We are currently working on further improving the optimization method for non-Poisson traffic. The key idea is to modify the cost function (4), using either simulation or the analytical model in [15], to better approximate the burst drop probability at a single link under a given non-Poisson arrival model. Then the network flow model in Section III-A can be used to compute a set of optimized routes for the specific arrival process at hand.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, N.J., 1993.

[2] I. Baldine *et al.* Just-in-time optical burst switching implementation in the ATDnet all-optical networking testbed. In *Proceedings of Globecom 2003*, volume 5, pages 2777–2781, San Francisco, USA, December 2003.

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

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

[5] R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, June 1961.

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

[7] R. Dutta and G. N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1(1):73–89, January 2000.

[8] B. Gluss. Further remarks on line segment curving-fitting using dynamic programming. *Communications of the ACM*, 5(8):441–443, August 1962.

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

[10] J. Iness *et al.*. Elimination of all-optical cycles in wavelength-routed optical networks. *Journal of Lightwave Technology*, 14(6):1207–1217, June 1996.

[11] H. Stone. Approximation of curves by line segments. *Mathematics of Computation*, 15(1):40–47, January 1961.

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

[13] X. Wang *et al.* Burst optical deflection routing protocol for wavelength routing WDM networks. In *Proceedings of Opticomm*, pages 120–129, 2003.

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

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

[16] A. Zalesky *et al.* 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.

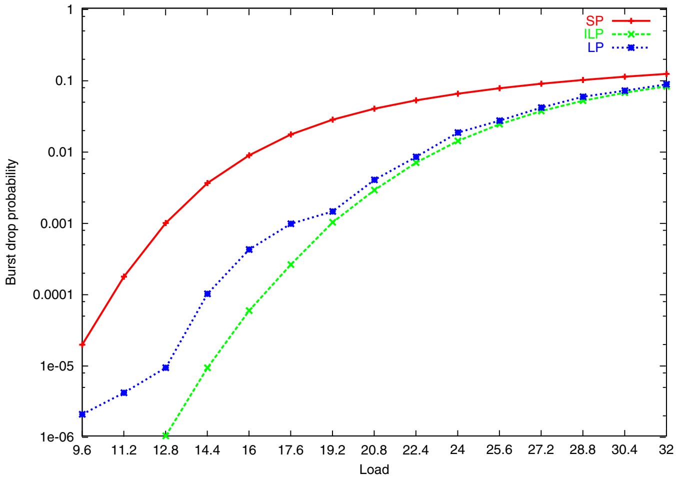

Fig. 8. Burst drop probability, Torus network, Poisson traffic, distance-dependent traffic pattern

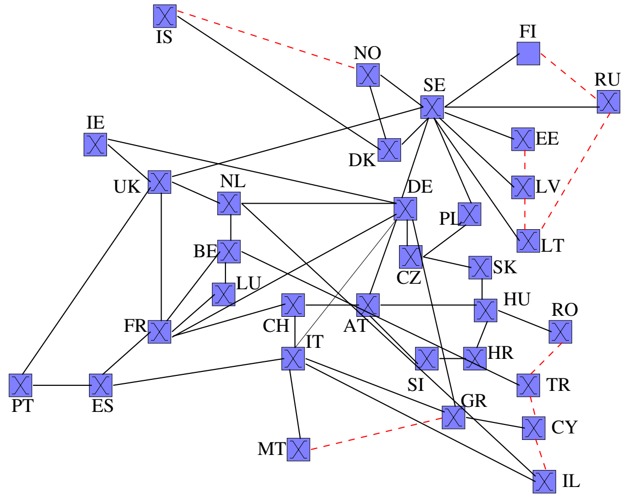

Fig. 11. The 33-node topology based on the 33-node GEANT network

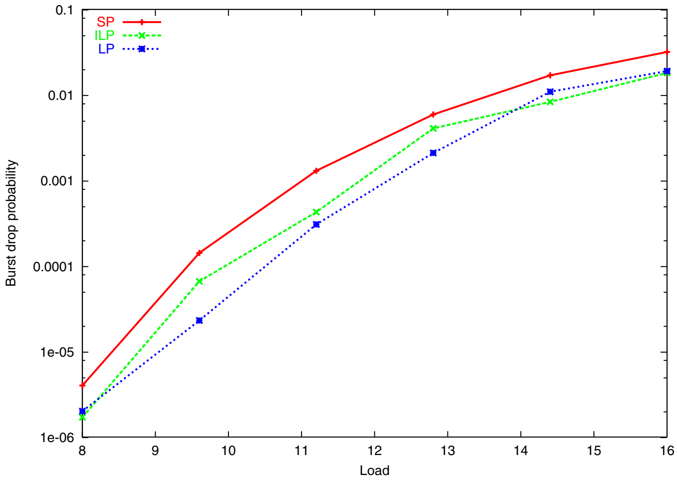

Fig. 9. Burst drop probability, NSF network, non-Poisson arrivals, distance-dependent traffic pattern

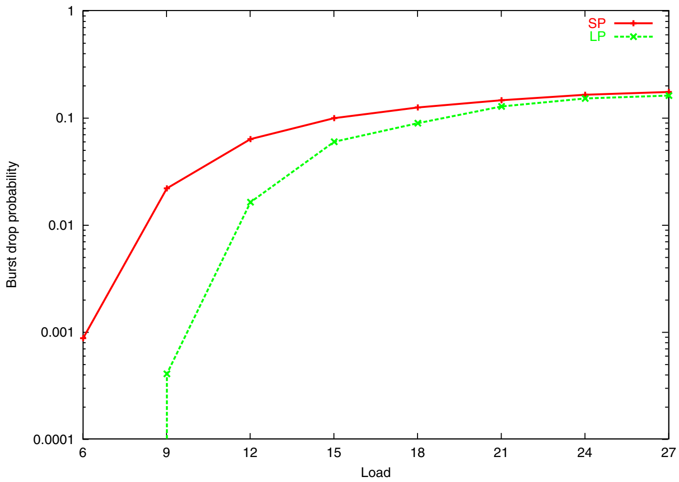

Fig. 12. Burst drop probability, GEANT network, Poisson traffic, distance-dependent traffic pattern

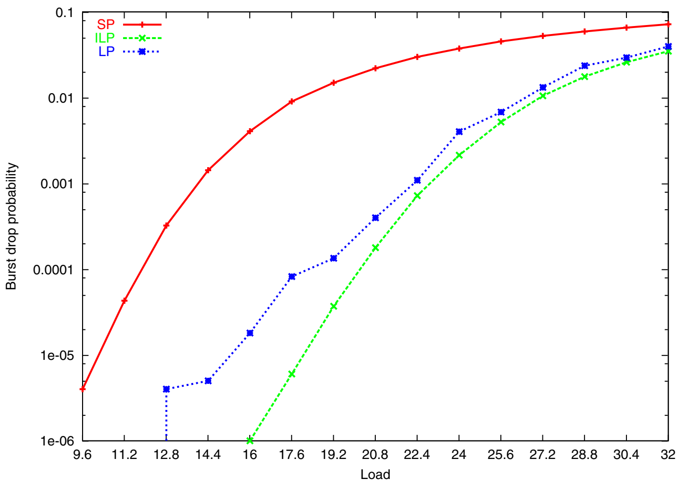

Fig. 10. Burst drop probability, Torus network, non-Poisson arrivals, distance-dependent traffic pattern

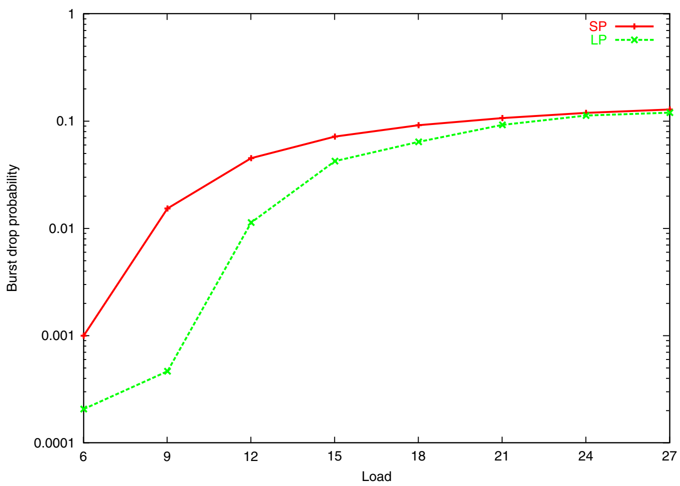

Fig. 13. Burst drop probability, GEANT network, non-Poisson traffic, distance-dependent traffic pattern