Routing Dependable Connections With Specified Failure Restoration Guarantees in WDM Networks

G. Mohan[a] and Arun K. Somani
Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011, USA
{gmohan, arun}@iastate.edu

Abstract— **This paper considers the problem of dynamically establishing dependable connections (D-connections) with specified failure restoration guarantees in wavelength-routed wavelength-division multiplexed (WDM) networks. We call a connection with fault tolerant requirements as a D-connection. We recommend using a pro-active approach to fault tolerance wherein a D-connection is identified with the establishment of a primary and a backup lightpath at the time of honoring the connection request. However, the backup lightpath may not be available to a connection throughout its existence. Upon occurrence of a fault, a failed connection is likely to find its backup path available with a certain specified guarantee.**

We develop algorithms to select routes and wavelengths to establish D-connections with specified failure restoration guarantees. The algorithms are based on a technique called *primary-backup multiplexing*. We present an efficient and computationally simple method to estimate the average number of connections per link for which the backup paths are not readily available upon occurrence of a link failure. This measure is used for selecting suitable primary and backup lightpaths for a connection. We conduct extensive simulation experiments to evaluate the effectiveness of the proposed algorithms on different networks. The results show that the blocking performance gain is attractive enough to allow some reduction in guarantee. In particular, under the light load conditions, more than 90% performance gain is achieved at the expense of less than 10% guarantee reduction.

I. Introduction

All-optical networks employing wavelength division multiplexing and wavelength routing are a promising candidate for future WANs. These networks offer the advantages of wavelength reuse and scalability. A *lightpath* is an 'optical communication path' between two nodes, established by allocating the same wavelength throughout the transmission path [1].

A lightpath is uniquely identified by a wavelength and a physical path.The requirement that the same wavelength must be used on all the links along the selected path is known as the *wavelength continuity constraint*. This constraint is unique to the WDM networks. Two lightpaths can use the same fiber link, only if they use different wavelengths. If two nodes are connected by a lightpath, a message can be sent from one node to the other without requiring any buffering and electro-optical conversion at the intermediate nodes. In other words, a message is transmitted in one (light)hop from the source to the destination. We assume that no wavelength converters are available at the routing nodes.

The problem of establishing lightpaths with the objective of minimizing the required number of wavelengths or minimizing the lightpath blocking probability for a fixed number of wavelengths is termed as the *lightpath establishment* problem (LE) [1]. For these, the establishment is either static (SLE), where a set of lightpaths is given a priori, or dynamic (DLE), where lightpaths are established and terminated on-the-fly [1]. A good routing and wavelength assignment (RWA) algorithm is important to improve the performance of wavelength-routed WDM networks. Several heuristic solutions for the RWA problem are available in the literature [1], [2], [3], [4]. The routing methods such as fixed and alternate path routing have been evaluated analytically and experimentally in [4], [5], [6].

Networks are prone to component failures. Therefore, providing fault tolerance capability to the connections is an important issue to be studied. We call a connection with fault tolerant requirements as a D-connection. In this paper, we focus on the problem of establishing D-connections with specified failure restoration guarantees in WDM networks with dynamic traffic demands. The objective is to improve the network blocking performance by allowing some reduction in restoration guarantee.

II. Dynamic Traffic and Failure Restoration

In a network with dynamic traffic demand, connection requests arrive to and depart from the network dynamically in a random manner. In response to new requests, lightpaths are established. A request may correspond to a single application and the entire lightpath bandwidth may be used exclusively by it. The dynamic traffic demand also results in several situations in transport networks as discussed in [7]. First, it may become necessary to reconfigure the network in response to changing traffic patterns. Second, with the rise in broadband traffic it is expected that the leased-line rates for private virtual networks and Internet service provider links will reach 2.5 Gb/s and higher. The demand for such services will change with time. Recently, there has been a growing interest in integrated IP/WDM routing [8]. In IP- over-WDM networks, a flexible virtual topology is used on the optical layer. The virtual topology changes frequently in response to the changes in the IP traffic patterns. The virtual topology is basically a set of lightpaths. In a flexible virtual topology, the connections on the optical layer (lightpaths) are short- lived. In [9], a distributed control protocol for routing lightpaths for realizing a flexible virtual topology to carry ATM traffic has been discussed.

We consider the single-link failure model in our study. We use a technique called *primary-backup multiplexing* for dynam-

This research was in part was supported by Nicholas Professorship at ISU, NSF grant number NCR-9796381, and NSF grant number ANI-9973102.

[a] The work was carried out when the author visited Iowa State University during January - June 1999.

ically establishing dependable connections with specified failure restoration guarantees combining the advantages of both the pro-active and reactive methods. When no connection is provided with a backup lightpath, the failure restoration guarantee is said to be zero. In this case, upon occurrence of a link failure, all the connections that use the failed link are terminated. If a failed connection between a node- pair needs to be established, then a new connection request is generated. If every connection is provided with a backup lightpath, then the failure restoration guarantee is said to be 100%. In case of the specified failure restoration guarantee, a failed connection will have its backup lightpath readily available with a certain guarantee, which could be less than 100%.

The motivation for our work is based on several facts. First, the faults do not occur frequently in practice to warrant full reservation. In such a case, reserving a backup lightpath (even with backup multiplexing) for every connection is wasteful and it leads to increased blocking of connection requests. Second, at any instant of time, the number of connections that require fault tolerance critically is very few. For such critical connections the backup lightpaths could be established (with no backup multiplexing) and for others the restoration guarantee could be less than 100%.

In this paper, we develop algorithms for routing dependable connections with specified failure restoration guarantees. We present an efficient and computationally simpler method to estimate the average number of connections per link which do not have their backups readily available upon occurrence of a single link failure. This measure will be used for selecting suitable primary and backup lightpaths for a connection. The proposed algorithms are flexible to choose a trade-off between the blocking performance improvement and guarantee reduction. We conduct extensive simulation experiments to evaluate the effectiveness of the proposed algorithms on different networks.

The rest of this paper is organized as follows. The related work on the fault tolerant routing problem are discussed in Section III .The backup multiplexing and primary-backup multiplexing techniques are explained with illustrations in Section IV. In Section V, the proposed estimator function is described. The proposed algorithms are discussed in Section VI. The results of the simulation experiments for various networks are discussed in Section VII. Finally, some concluding remarks are made in Section VIII.

III. Related Work

The fault-tolerant routing problem for the dynamic traffic has been earlier addressed for non-WDM networks such as ATM networks. Some of the existing approaches for fault-tolerant routing have been surveyed and a pre-routing scheme based on *backup multiplexing* has been proposed in [10]. When a link or node fails, all the connections currently using this link or node fail. The methods for recovering from the failure can be broadly classified into reactive and pro-active methods. The reactive method is the simplest way of recovering from failures. In this method, when an existing connection fails, a new connection which does not use the failed components is selected and established if available. This has an advantage of low overhead in the absence of failures. However, this does not guarantee successful recovery, as the attempt to establish a new connection may fail due to resource shortage at the time of failure recovery. Also, in case of distributed implementation, contention among simultaneous recovery attempts for different failed connections may require several retries to succeed, resulting in increased network traffic and service resumption time.

To overcome the above difficulties, pro-active methods can be employed. In the *end-to-end detouring* pro-active method, a backup connection is established between two end nodes of a primary connection. The backup connection takes over the role of the primary connection when the primary connection fails. Each backup connection reserves its own spare resources, so that there will be no conflict among multiple path recovery attempts. Since the backup connection is established before the failures actually occur, one can use it immediately upon occurrence of a failure in the primary, without invoking the time-consuming connection re-establishment process. Hence, the failure recovery delay of this pro-active method is much smaller leading to fast recovery. However, this method reserves excessive resources.

In [10], a resource sharing technique, called *backup multiplexing* has been proposed to minimize the spare resources required on a link. It reserves only a small fraction of link-resources needed for all backup connections traversing the link. This method is used to establish dependable connections in an efficient way in terms of amount of spare resources. A dependable connection (a D-connection for short) consists of a primary connection and one or more backup connections. Each backup connection remains as a cold standby until it is activated. The idea behind the backup multiplexing is that two backup connections can share the resource on a link, if their corresponding primary connections do not fail simultaneously. This happens when they do not share any link, for a single link failure model.

In [11], some mechanisms to detect and isolate faults such as fiber cuts and router failures have been presented. The problem of fault tolerant design of WDM networks has been addressed in [12], [13] for static traffic demand. Here, a set of connection requests is given a-priori and lightpaths are assigned for them. For every active lightpath, a set of backup lightpaths is predetermined to handle all possible fault occurrences. The objective of these design algorithms is to minimize the required spare resources such as wavelengths and fibers in order to incorporate fault tolerance. These algorithms can afford to be computationally expensive as they are run off-line. On the other hand, the dynamic routing schemes must use simpler and faster algorithms because, in a dynamic traffic environment, short-lived connections are setup and torn down frequently. Some dynamic algorithms for fault-tolerant routing in WDM networks have been recently proposed in [9], [14]. These algorithms use distributed protocols to find routes avoiding the faulty components. Basically, these algorithms are 'reactive' in nature and find a new route after the occurrence of component failures.

IV. Multiplexing Techniques

In this section, we describe the backup multiplexing and primary-backup multiplexing techniques with illustrations.

A. Definitions

We now define terms that are needed to describe the multiplexing techniques and also the proposed algorithms.

Definition 1: A link refers to a fiber in the network. A wavelength channel (also called a channel) refers to a wavelength on a link. A physical path consists of a sequence of links. A lightpath consists of a sequence of channels with all the channels using the same wavelength.

Definition 2: The path vector P of a lightpath defines the set of links used by the lightpath. If L is the number of links in the network then P is an L-bit vector, $< p_0 p_1 ... p_{L-1} >$ and the bit value of 1 in position i means that link i is used by the lightpath.

Definition 3: The conflict vector C of a channel defines the set of links used by the primary lightpaths that correspond to the backup lightpaths multiplexed onto the channel. If L is the number of links in the network then C is an L-bit vector, $< c_0 c_1 ... c_{L-1} >$ and the bit value of 1 in position i means that link i is used by a primary lightpath that corresponds to some backup lightpath traversing the channel.

Definition 4: A channel is said to be dirty if a primary lightpath and one or more backup lightpaths use it. Otherwise, it is said to be pure.

Definition 5: A connection is called an orphan if its backup lightpath is not free, i.e. it traverses one or more dirty channels. A connection that is not an orphan is said to be safe.

Definition 6: A channel is said to be weak if it is used by the primary lightpath of an orphan.

Definition 7: The weak channels induced by a dirty channel are the channels used by those primary lightpaths whose backup lightpaths use the dirty channel.

Definition 8: The weak channels induced by a lightpath are the set of all the distinct weak channels induced by the dirty channels on the lightpath.

B. Backup Multiplexing

In order to use the wavelength channel resources efficiently, the backup multiplexing technique can be used. Two backup lightpaths can share a channel if the corresponding primary lightpaths do not fail simultaneously. We consider the single-link failure model. Therefore, if two primary lightpaths are link- disjoint, their backup lightpaths can be multiplexed onto the same channel. We illustrate this technique with an example. Consider a network with 5 nodes and two wavelengths w_0 and w_1 as shown in Fig. 1. The figure is a layered graph representation of the network with two wavelength layers w_0 and w_1. The figure shows three pairs of lightpaths, $< p_1, b_1 >$, $< p_2, b_2 >$, and $< p_3, b_3 >$ where b_i is the backup path of the primary path p_i. While the paths p_1 and p_2 use the wavelength w_0, the paths p_3, b_1, b_2, and b_3 use the wavelength w_1. All the three primary lightpaths are link- disjoint and any single link failure will fail at most one of them. Therefore, their backup lightpaths can share any edge. For example, the edge $0 \rightarrow 2$ on the wavelength w_1 is shared by all the three backup lightpaths. The conflict vector associated with the channel $0 \rightarrow 2$ on w_1 has 1's in positions that correspond to the links, 0-1, 1-4, 0-3, and 4-2. This means that any connection whose primary lightpath uses any of these links can not use the channel $0 \rightarrow 2$ on w_1 for its backup.

C. Primary-Backup Multiplexing

The primary-backup multiplexing technique allows a primary lightpath and one or more backup lightpaths to share a channel. By using this technique, increased number of connection requests can be satisfied at the expense of reduction in failure restoration guarantee. A connection loses its backup lightpath and becomes an orphan when a channel assigned to its backup lightpath is used by some other primary lightpath. In a dynamic traffic scenario, an orphan can again become safe when the primary lightpaths using some channels of an orphan's backup path terminate. A connection loses its recoverability only when the following three events occur simultaneously. 1) a link fails 2) the failed link is used by its primary lightpath, and 3) the connection is an orphan. However, such a situation is less probable. Also, except under the heavy load conditions, upon occurrence of a link failure, an orphan connection may be able to find a new backup lightpath, if the search process is invoked.

In order to maximize the recoverability of connections, the primary-backup technique may be used only if routing a new D-connection fails. If the primary lightpath of a new connection is routed on a channel that is being shared by backup lightpaths of some existing connections, then those connections become orphans. If the backup lightpath of a new connection is routed on a channel that is being used by a primary lightpath of an existing connection, then the new connection becomes an orphan.

We illustrate this technique with an example. Consider a network with five nodes and two wavelengths per fiber as shown in Fig. 2. The figure shows four pairs of lightpaths. Assume that initially, the connections $< p_1, b_1 >$, $< p_2, b_2 >$, and $< p_3, b_3 >$ are routed. When a new request arrives for a connection from node 1 to node 2, there does not exist a free route for a primary-backup lightpath pair. This request is rejected if no primary-backup multiplexing is used. By using the primary-backup multiplexing technique, this request can be honored by allocating the pair $< p_4, b_4 >$ as shown in the figure. The primary lightpath p_4 traverses channels used by the backup lightpath b_3 and hence the connection $< p_3, b_3 >$ becomes an orphan. In this case, the channels $1 \rightarrow 4$ and $4 \rightarrow 2$ on w_1 are dirty. Alternatively, the new connection request can be honored by allocating the pair $< p_4, b_4 >$ as shown in Fig. 3. Here, the backup lightpath b_4 uses a channel used by the primary lightpath p_1 and hence the new connection is also an orphan. In this case, the channel $1 \rightarrow 4$ on w_0 is dirty. It can be noted that b_4 can not be routed on w_1 as p_4 and p_3 are not link-disjoint and because of this reason, b_4 and b_3 can not be multiplexed on the

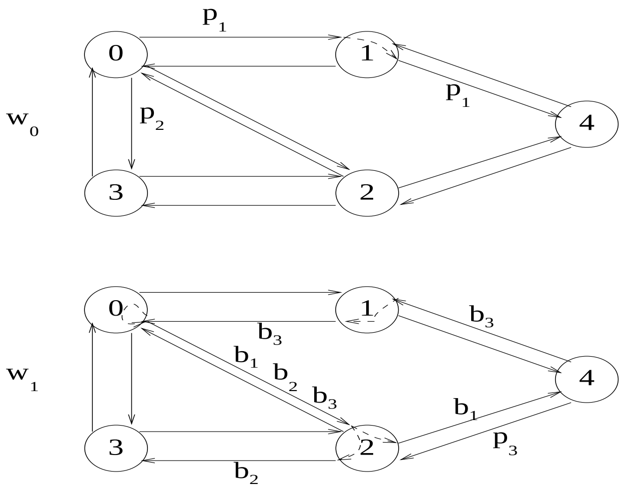

Fig. 1. Illustration of the Backup Multiplexing technique.

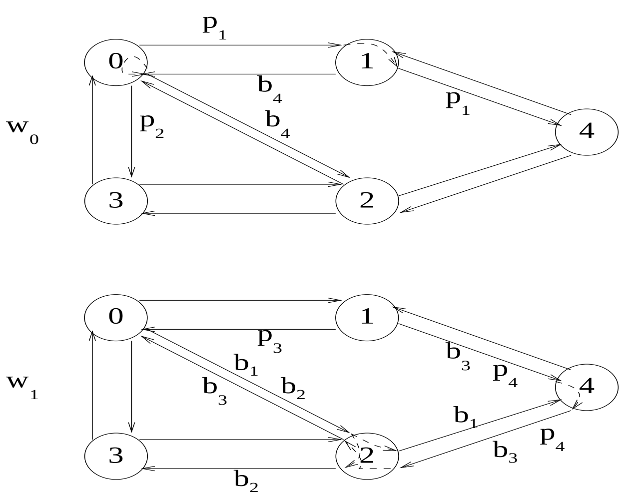

Fig. 2. Illustration of the Primary-Backup Multiplexing technique: Case-1.

links $1 \rightarrow 4$ and $4 \rightarrow 2$ on w_1.

V. Computation of the Number of Orphans

When a backup lightpath of a connection is routed on a channel that is being used by some other primary lightpath, the connection itself becomes an orphan. This means that on every link used by the primary lightpath of the connection, a weak channel (and also an orphan) is created. When a primary lightpath is routed on a channel carrying a set of backup lightpaths, the connections that require these backup lightpaths become orphans. Our goal is to minimize the number of orphans on a link so as to maximize the failure restoration guarantee. To do so, we need to know the effect of setting up a new primary path on the number of orphans. Let B_p be the set of distinct backup lightpaths that use some edges of a primary lightpath L_p. The sum of hop lengths of the primary lightpaths that correspond to the backup lightpaths from the set B_p gives the required number of weak channels induced by L_p.

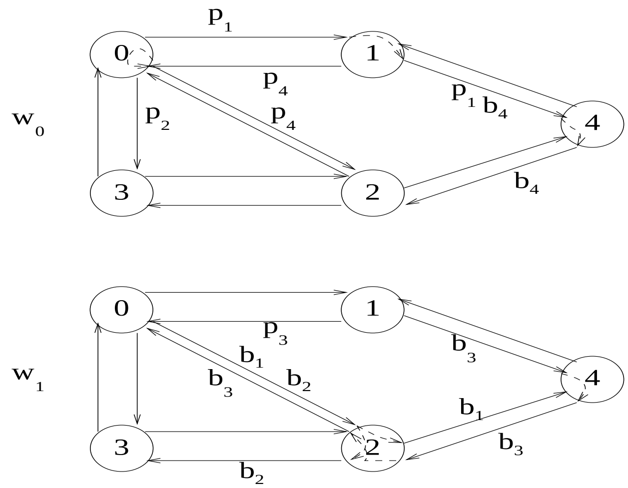

Fig. 3. Illustration of the Primary-Backup Multiplexing technique: Case-2.

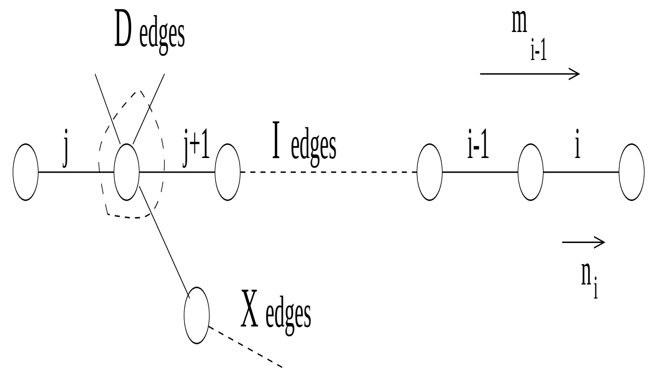

Fig. 4. Estimating the number of weak channels induced by a path.

A. Need for an Estimator Function

In order to compute the number of weak channels induced by L_p, we need to keep the identity of all the backup lightpaths that are multiplexed onto a channel. Keeping track of identity of the backup lightpaths on various channels and finding the set of distinct backup lightpaths B_p is computationally expensive. However, keeping track of the number of backup lightpaths and the sum of the hop-count of their corresponding primary lightpaths can be done with simple operations. However, this has a shortcoming. Since the same backup lightpath can traverse more than one edge of L_p, counting the number of distinct backup lightpaths in B_p and also the number of channels used by their corresponding primary lightpaths is not trivial. Therefore, we need to make a compromise between the complexity and the accuracy of computing the number of induced weak channels. A computationally simpler heuristic function, which can estimate the number of induced weak channels accurately, is highly desirable.

We develop a computationally faster heuristic method to make an estimation of the number of weak channels induce when a primary lightpath L_p is established. This can be used to estimate the average number of orphans created per link. The proposed estimator function uses $O(H)$ number of operations, where H is the hop length of L_p.

B. The Estimator Function

We now describe a method to estimate the number of distinct backup lightpaths in B_p, denoted by $N(B_p)$. We also explain how the same method can also be used to estimate the number of weak channels induced by L_p by keeping track of the count of the number of edges of the primary lightpaths that correspond to the backup lightpaths that are multiplexed onto a channel.

Consider a primary lightpath L_p whose edges include j, $j+1$, $i-1$, and i as shown in Fig. 4

Let n_i be the number of backup lightpaths multiplexed onto edge i.

Let m_i be the number of backup paths that use edge i and continue to the next edge $i+1$.

Let e_i, $1 \leq i \leq H-1$ be the number of backup lightpaths that use edge i and does not use any of the edges 0 through $i-1$. The value of e_0 is n_0. The value of e_1 is $n_1 - m_0$.

The value of $N(B_p)$ is computed as $N(B_p) = \sum_{i=0}^{H-1} e_i$

Computation of e_i

The value of e_i for edge i can be estimated as follows.

Let t_i be the number of backup paths that use some edge $j < i$, use edge i, but do not use any of the edges $j+1, j+2, ..., i-1$.

The value of e_i is computed as $e_i = n_i - t_i - m_{i-1}$.

Computation of t_i

The value of t_i is computed as follows.

Let k_j be the number of backup paths that use edge j, but do not continue to the next edge $j+1$. Therefore, $k_j = n_j - m_j$.

Let $k_{j,i}$ be the number of backup paths that use edge j and use edge i without using any of the edges $j+1, j+2, ..., i-1$.

We now explain how the value of $k_{j,i}$ can be computed.

Let D be the degree of the end vertex v of edge j, I be the number of I intermediate edges on the path L_p between and including $j+2$ and $i-2$, and L be the total number of links in the network. Let an arbitrary path p among the k_j paths traverses X edges excluding the edges incident on v. We assume that, other than the D edges incident on v and the I intermediate nodes, every edge is equally likely to be traversed by it.Although this assumption depends on the factors such as the topology of the network and the choice of alternative routes for the node-pairs, it is not unrealistic as the value and range of I is small for practical networks, and also among D edges incident on v, one or two are traversed by p and the other edges cannot be traversed by it.On such a path p, edge i appears with the probability $r = \frac{X}{L-(D+I)}$.

If q_i denotes the probability that a backup path on edge i does not enter from edge $i-1$, then it is calculated as $q_i = \frac{n_i - m_{i-1}}{n_i}$, for $n_i \neq 0$, otherwise $q_i = 0$. The probability that the path p enters edge i other than edge $i-1$ is then given by $r \times q_i$.

The value of $k_{j,i}$ is then computed as $k_{j,i} = k_j \times r \times q_i$

We choose appropriate values for X, D and I depending on the topology and hop-counts of the alternative routes and make r a constant. An appropriate choice of these values will make the computation of $k_{j,i}$ and t_i simpler. The value of D and I depends on j for any given value of i. However, the range of possible values of D and I is small and also the deviation of the assumed value of $D+I$ from the actual value is not significantly comparable with L. Therefore, choosing a constant value for r does notintroduce any significant error. Our experimental results presented in Section V confirm it. Note that the estimator function is a heuristic only. Instead of considering r as a constant, we can compute it for every possible value of j for a given i. But, this increases the complexity of the estimator function.

The prefix sum S_i is defined and computed as

$$S_i = k_0 + k_1 + ... + k_i.$$

The value of t_i is then computed as

$$t_i = \sum_{j=0}^{i-2} k_{j,i} = r \times q_i \times S_{i-2}$$

Computing the Number of Orphans

The above estimator function can also be used to determine the number of weak channels created by routing a primary lightpath L_p with slight modificationss. Let n_i be used to denot the number of weak channels induced by edge i. This value is nothing but the total number of channels used by those primary lightpaths whose backup lightpaths are multiplexed onto edge i. Let m_i be used to denote the number of common weak channels induced by edge i and edge $i+1$. Let the other variables be changed accordingly. If m is the number of weak channels induced by L_p and every link is equally likely to be the failed link, then the average number of orphans created per link by establishing L_p is given by m/L, where L is the total number of links in the network.

function *Estimate*(L_p)
Given a path L_p with H edges and their n and m values.
This procedure estimates the number of weak channels induced by L_p.
step 1: (* process edges 0 and 1 *)
    $e_0 \leftarrow n_0$; $e_1 \leftarrow n_1 - m_0$; $val \leftarrow e_0 + e_1$
    $k_0 \leftarrow n_0 - m_0$; $k_1 \leftarrow n_1 - m_1$;
    $S_0 \leftarrow k_0$; $S_1 \leftarrow k_0 + k_1$;
step 2: (* process edges 2 through $H-1$ *)
    For i=2 to $H-1$ do
    begin
        If $n_i = 0$ then $q_i \leftarrow 0$; else $q_i \leftarrow \frac{n_i - m_{i-1}}{n_i}$
        $t_i \leftarrow r \times q_i \times S_{i-2}$
        $e_i \leftarrow n_i - t_i - m_{i-1}$
        $val \leftarrow val + e_i$
        $k_i \leftarrow n_i - m_i$
        $S_i \leftarrow S_i + k_i$
    end
step 3: Return(val)

VI. Proposed Algorithms

We now describe the key idea and working of the proposed algorithms, *Limited_Average_Orphans (LAO)* and *Limited_Orphans (LO)*. These algorithms use backup multiplexing for efficiently using the wavelength channels and use primary-backup multiplexing for improving the blocking performance with specified failure restoration guarantees. The key idea behind the *LAO* algorithm is to ensure certain restoration guarantee by limiting the average number of orphans created per link upon occurrence of any single link failure to a predefined value. The *LO* algorithm ensures this by limiting the number of orphans on any link to a predefined value. The algorithms basically use alternate routing method. For every pair of source and destination, a set of alternative routes (also referred to as candidate routes) is pre-computed off-line. The candidate routes for a source-destination pair are chosen to be link-disjoint to incorporate fault tolerance.

A. *Description of the* LAO *algorithm*

When a new request arrives for a D-connection between a source-destination pair $< s, d >$, a primary-backup lightpath-pair $< L_p, L_b >$ is to be chosen to satisfy the request. It is chosen in such a way that it is admissible and its cost is minimal. We say that the network state is safe, if the average number of orphans per link does not exceed a predefined orphan threshold value T. A lightpath pair is said to be admissible, if its establishment does not take the network into an unsafe state. This algorithm has two components: cost computation and admissibility test.

Cost Function

The cost of the primary lightpath L_p, denoted by $C_p(L_p)$, is the number of free channels used by it. It is to be noted that the channels used by it are either free or used by some backup lightpaths. If any channel is used by some other primary lightpath, the cost becomes infinity. If a channel carries some backup lightpaths, it becomes dirty.

The cost of a backup lightpath L_b for a given primary lightpath L_p, denoted by $C_b(L_b, L_p)$ is defined as the number of free channels used by it. If a channel is free, then the cost of using it is 1. If a channel is currently used only by a primary lightpath, then the channel becomes dirty. If a channel is currently used by a set of backup lightpaths S, then it can be used by L_b with no extra cost, if and only if its primary route is link-disjoint with the primary route of each and every backup lightpath in the set S. In other words, the bit-AND operation of the conflict vector of the channel and the path vector of the L_p should yield 0 in order for L_b to use the channel. The cost of a D-connection using the primary-backup lightpath pair $< L_p, L_b >$ is given by

$C_D(L_p, L_b)$ = $C_p(L_p)$ + $C_b(L_b, L_p)$ + PenaltyCost*N_d,

where PenaltyCost is the cost of a dirty channel and N_d is the number of dirty channels on both the primary and backup lightpaths. The value chosen for PenaltyCost is such that it is larger than the cost of any lightpath-pair with no dirty edges.

function *Cost*(L_p, L_b)
This function computes the cost of a lightpath-pair $< L_p, L_b >$.
step 1: (* Compute the cost of L_p *)
  $C_p \leftarrow 0$; $N_d \leftarrow 0$
  For every edge i of L_p do
  begin
    If i is free
    then $C_p \leftarrow C_p + 1$
    else if i is used by a primary lightpath
      then $C_p \leftarrow \infty$
    If i is used by a backup lightpath
    then $N_d \leftarrow N_d + 1$ (* a dirty edge is used *)
  end
step 2: (* Compute the cost of L_b *)
  $C_b \leftarrow 0$; $P \leftarrow$ Path vector of L_p
  For every edge i of L_b do
  begin
    $C \leftarrow$ Conflict vector of i
    If i is free
    then $C_b \leftarrow C_b + 1$
    else if P bitAND C $\neq 0$ then $C_b \leftarrow \infty$
    If i is used by a primary lightpath
    then $N_d \leftarrow N_d + 1$ (* a dirty edge is used *)
  end
step 3: (* Return the cost of the lightpath pair *)
  Return ($C_p + C_b + Penalty - Cost \times N_d$).

Admissibility Test

The admissibility test for a pair $< L_p, L_b >$ is performed as follows. The algorithm keeps track of the values of the number of weak channels induced by a channel and the number of weak channels in common among those induced by a channel and the next channel. This information is updated whenever a backup lightpath is established and released. The updation requires only a constant number of operations for a channel on the backup lightpath. It computes the approximate value of the number of orphans per link, denoted by L_{orp}. Initially the value of L_{orp} is zero. Let x and y be the number of orphans created per link by L_p and L_b, respectively. If L_p traverses at least one dirty channel, then the number of weak channels induced by L_p, m is computed using the estimator function. The value of x is then computed as $\frac{m}{L}$. If the backup lightpath L_b has at least one dirty channel, then the weak channels induced by it are nothing, but the channels used by L_p. If h is the hop length of L_p then the value of y is computed as $\frac{h}{L}$. The pair is admissible if $L_{orp} + x + y$ does not exceed the orphan threshold value T.

The choice of a value for the orphan threshold parameter T has an effect on the acceptance rate and failure restoration guarantees of connections. A low value for T will result in lower acceptance rate and higher restoration guarantee. On the other hand, a high value for T will result in higher acceptance rate and lower restoration guarantee. Therefore, we can achieve a desired tradeoff between the network performance and restoration guarantee by choosing an appropriate value for T.

function *AdmissibilityTest*(L_p, L_b, T)
This function checks if the pair $< L_p, L_b >$ is admissible.
The orphan threshold value is T.
step 1: $m \leftarrow 0$; $h \leftarrow 0$
step 2: If L_p traverses a dirty channel
then $m \leftarrow$ *Estimate* (L_p)
step 3: If L_b traverses a dirty channel
then $h \leftarrow$ *Hop_length* (L_p)
step 4: If $T \geq L_{orp} + \frac{m+h}{L}$
then Return (*Success*)
else Return (*Fail*)

The algorithm chooses the minimum cost pair among those, which is admissible. Once the pair is established, the value of L_{orp} is updated by adding $x + y$ to it. When, L_p is released, the number of orphans per link induced by it (say x') is calculated and L_{orp} is updated by subtracting x' from it. It is to be noted that the computation of x for L_p does not require the global knowledge, i.e. it is computed independent of other existing connections. It may so happen that L_p uses a channel that is used by a backup lightpath L_b^j of some connection which was already made orphan by some other lightpath L_p^1. In that case, orphan count is redundantly updated by L_p. This would introduce an error in estimating L_{orp}. However, this error is corrected when L_p is released before L_b^j is released, as a similar situation arises. If L_b^j is released first, then the error gets corrected by the following updation. When a backup lightpath is released, the number of distinct primary lightpaths traversed by it (say n') is computed using the estimator function with a suitable definition of the variables used. This backup lightpath would have been counted by the n' number of primary lightpaths, and hence the value of L_{orp} is updated by subtracting $\frac{n' \times h}{L}$ from it. Here, h is the hop length of the primary lightpath that corresponds to the backup lightpath to be released.

The following features make the *LAO* algorithm attractive:

1. The estimator function is computationally simpler. This makes the algorithm faster and suitable for dynamic routing.
2. The algorithm does not require global knowledge of network state information like how the existing connections are routed, the state of the channels and the identity of the backup lightpaths that are multiplexed onto the channel. This makes the algorithm suitable for distributed implementation.
3. The algorithm is flexible to choose a desired tradeoff between the network performance and the failure restoration guarantee.

B. *Description of the* LO *algorithm*

The *LO* algorithm also chooses the minimum cost lightpath pair among all the admissible pairs to satisfy a new connection request. The cost of a lightpath pair is calculated in the same way as the *LAO* algorithm. However, the admissibility criterion is different from that used by the *LAO* algorithm. A lightpath pair is said to be admissible by this algorithm, if establishing it does not result in the number of orphans on any link exceeding the orphan threshold T. While the *LAO* algorithm limits the average number of orphans per link to T, the *LO* algorithm limits the number of orphans on any link to T. So, this algorithm guarantees that the number of orphans created upon occurrence of any link failure is at most T.

It does not make any estimation of the average number of orphans per link. Instead it computes the actual number of orphans on any link. This is possible, as this algorithm keeps track of the orphans on every link. Also, it keeps the identity of the connections whose backup lightpaths are multiplexed onto a channel. It is therefore computationally more complex. It is also less amenable for distributed implementation as it requires global network state information.

Admissibility Test

To decide the admissibility of a lightpath pair $< L_p, L_b >$, the following steps are followed.

1. Determine the set of connections whose backup lightpaths use some dirty channel(s) on L_p. Call this set as S. If L_b has any dirty edge then add the new connection $< L_p, L_b >$ to the set.
2. Let S_l be the set of links used by the primary lightpaths of the connections in S.
3. Temporarily transform the network state into a new state by marking as weak, the channels that are used by the primary lightpaths of the connections in the set S.
4. If the the number of weak channels on every link from the set S_l does not exceed the orphan threshold T, then the pair $< L_p, L_b >$ is admissible. Otherwise, it is not admissible.

VII. Performance Study

We evaluate the effectiveness of the proposed algorithms by extensive simulation. The simulation networks considered are the 21-node ARPA-2 network with 26 duplex links aand 16-node Mesh-torus networ with 32 duplex links. A duplex link is comprised of two simplex links in the opposite directions. Every simplex link is assumed to have 8 wavelengths and therefore a duplex channel consists of 16 wavelength channels.

The values chosen for X and $D + I$ for the Mesh-torus network are 3 and 5, respectively, and for the ARPA-2 network, 4 and 6, respectively. The connection requests arrive at a node as a Poisson process with exponentially distributed holding time with unit mean. Every node is equally likely to be a destination node for a connection request.

Performance Metric

We use two metric, **relative performance gain** and **reduction in guarantee**, to measure the performance of the proposed algorithms. We compare the performance of the proposed algorithm with that of the zero-percent-guarantee algorithm and the 100-percent guarantee algorithm. The zero-percent-guarantee algorithm does not provide any backup lightpath for the connections. The 100-percent-guarantee algorithm provides backup lightpath for every connection. While it uses backup multiplexing for improving the blocking performance, it does not use primary-backup multiplexing.

Let b_0 be the connection blocking probability of the zero-percent-guarantee algorithm, b_{100} be the connection blocking probability of the 100-percent-guarantee algorithm, and b be the connection blocking probability of the proposed algorithm. The relative performance gain is calculated as $\frac{b_{100}-b}{b_{100}-b_0}$. The reduction in guarantee is defined as the probability that a connection does not find its backup lightpath readily available upon occurrence of a link failure.

Numerical Results

We plot the percentage of performance gain and guarantee reduction as a function of orphan threshold T. Theoretically, the value of T can go upto 16. In practice, the number of orphans on an individual link may be as high as 16, but the average number of orphans per link may be less. This is reflected in the performance curves. The curves in case of the LAO algorithm level off for a smaller value of T when compared to those in case of the LO algorithm.

The performance of the *LAO* algorithm is plotted in Fig. 5 for the Mesh-torus network. The relative performance gain and reduction in guarantee for three different arrival rates per node (r), 4,7, and 10 are plotted as a function of the orphan threshold T. The chosen arrival rates reflected the light, medium, and heavy traffic load conditions of the network. For the 100-percent-guarantee algorithm (with backup multiplexing), these arrival rates correspond to the blocking probability of 0.024, 0.224, and 0.372, respectively.

The curves show that under light load conditions, more than 90% performance gain can be achieved at the expense of less than 10% reduction in guarantee. This is because, when the load is light, the number of backup lightpaths multiplexed on a dirty channel is less and the shorter-hop routes are more likely to be used by the primary lightpaths. Therefore, at any instant of time, the number of weak channels per link is very less and also the number of orphan connections in the network is also very less. As the traffic loading increases, the gap between these two metric decreases. However, even for the heavy load condition, the performance gain is more when compared to the reduction in guarantee. This demonstrates the usefulness of the algorithm.

To evaluate how good and accurate the estimator function is, we plot the performance of the algorithm *LAO-actual* in Fig. 6. This algorithm is the same as the *LAO* algorithm except that the average number of orphans per link is actually measured instead of estimated. It is observed that the estimator function is accurate except when the traffic load is high and the threshold value is small. The reason for this can be explained as follows. When the threshold value is less, the error introduced by the estimator function is comparable to the threshold value, consequently, the result of admissibility test goes wrong. Also, if the load is high, the number of connections arrived during the period of error will be high. However, the error does not diminish the usefulness of the algorithm based on estimator function.

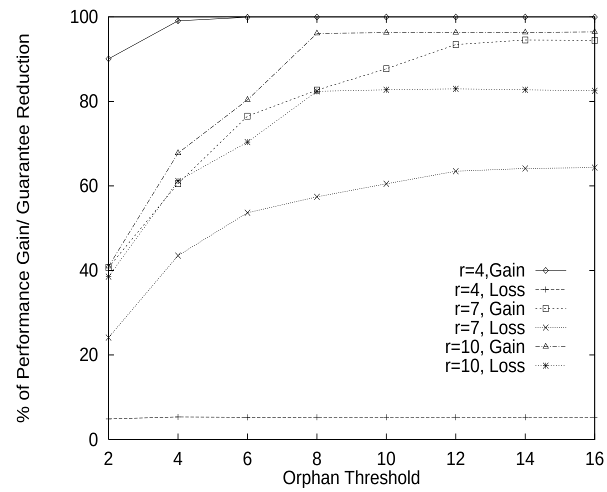

Fig. 5. Performance of the *LAO* algorithm under different loading conditions for the Mesh-torus network.

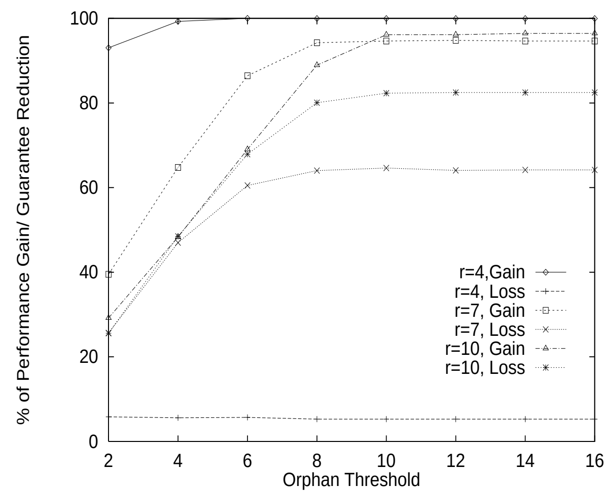

Fig. 6. Performance of the *LAO-actual* algorithm under different loading conditions for the Mesh-torus network.

The performance of the *LO* algorithm is plotted in Fig. 7 for the Mesh-torus network. We observe that the curves change slowly before level off, when compared to that of the *LAO* algorithm. The reason is as follows. The *LO* algorithm limits the number of orphans on every link whereas the *LAO* algorithm limits the average number of orphans on a link. Therefore, the *LO* algorithm is more restrictive than the other one and hence the the rate of change of curves is slower.

The performances of the *LAO*, *LAO-actual*, and *LO* algorithms for the ARPA-2 network are shown in Fig. 8, 9,and 10, respectively. The relative performance gain and reduction in guarantee for three different arrival rates per node (r), 0.75, 1.50, and 2.25 are plotted as a function of orphan threshold T. The chosen arrival rates reflected the light, medium, and heavy traffic load condition of the network. For the 100-percent-guarantee algorithm (with backup multiplexing), these arrival rates correspond to the blocking probability of 0.052, 0.244, and 0.389, respectively. The results demonstrate the usefulness

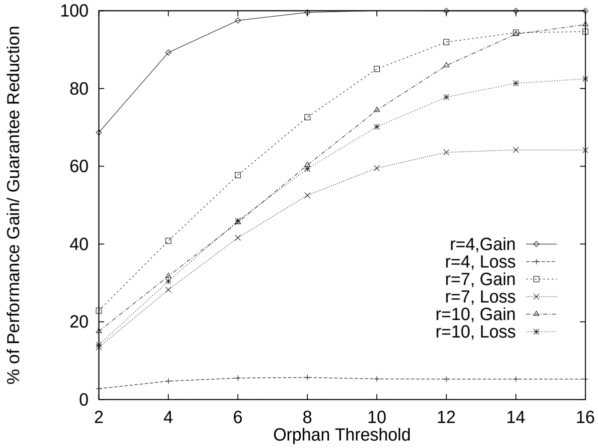

Fig. 7. Performance of the *LO* algorithm under different loading conditions for the Mesh-torus network.

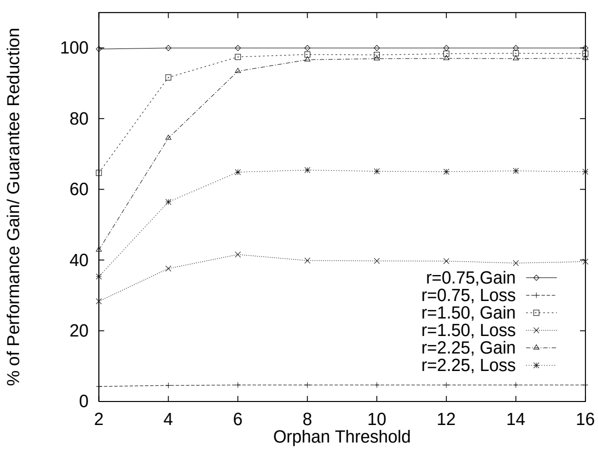

Fig. 8. Performance of the *LAO* algorithm under different loading conditions for the ARPA-2 network.

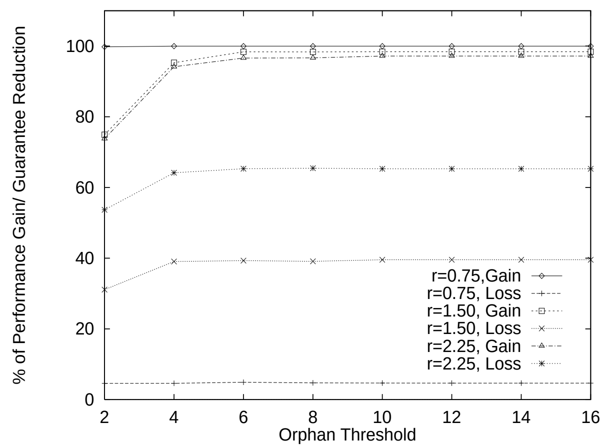

Fig. 9. Performance of the *LAO-actual* algorithm under different loading conditions for the ARPA-2 network.

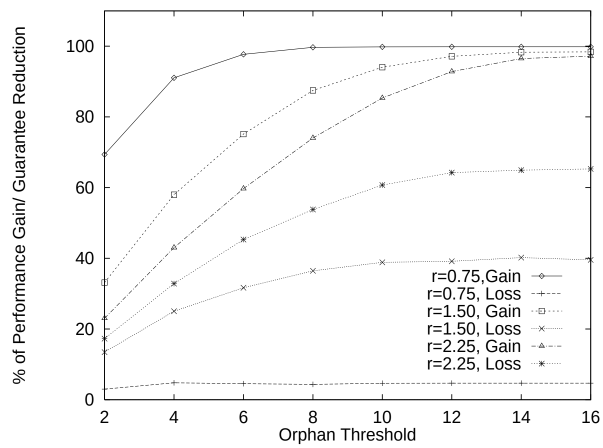

Fig. 10. Performance of the *LO* algorithm under different loading conditions for the ARPA-2 network.

of the proposed algorithms. We notice that the performance gain is more for the ARPA-2 network when compared to the Mesh-torus network. The reason is as follows. The ARPA-2 is a sparsely connected network. The connections are longer and the number of possible link-disjoint connections for any source-destination pair is less. Therefore, the usefulness of a mere backup multiplexing is less in sparsely connected networks and using the primary-backup multiplexing technique will result in acceptance of increased number of connections.

Fig. 11 and 12 depict the average number of orphans created per link by the *LAO* and *LAO-actual* algorithms vs the orphan threshold parameter (T) for different traffic load conditions for the Mesh-torus and ARPA-2 networks, respectively. Fig. 13 and 14 depict the average number of orphans created per link by the *LO* algorithm vs the orphan threshold parameter (T) for different traffic load conditions for the Mesh-torus and ARPA-2 networks, respectively. From these plots, we observe that when the load is light, the number of orphans created per link is very low and less than one. As the load increases, the number of orphans per link also increases. However, the number of orphans level off indicating that maximum loss is bounded. This is because although a duplex link has 16 channels, not all of them are used by primary lightpaths. We also notice that the change is slow for Algorithm *LO* as it is more restrictive and limits the number of orphans created on the links individually.

VIII. Conclusions

In this paper, we have addressed the problem of dynamically establishing primary-backup lightpaths for dependable connections in wavelength-routed WDM networks with specified failure restoration guarantees. We developed different algorithms, which are based on a new technique called *primary-backup multiplexing.* The key idea of our algorithms is to limit the number of connections that will not have their backups readily available when a fault occurs to a pre-defined threshold value. We define such connections as orphans and consider only a single link failure. To estimate the number of orphans at the time of establishing a connection, we developed an *estimator* function. This function is computationally simple and does not require any global knowledge of the Network State. The effectiveness of the algorithms have been evaluated using extensive

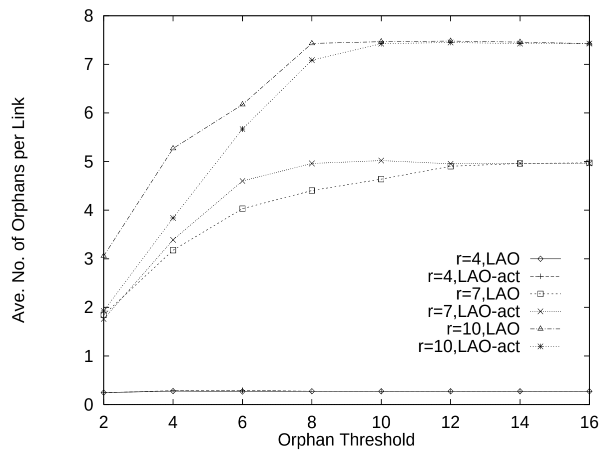

Fig. 11. Average number of orphans created per link by the *LAO* and *LAO-actual* algorithms for the Mesh-torus network.

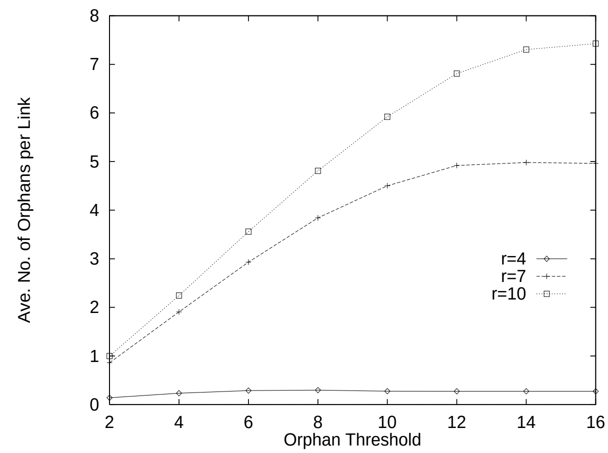

Fig. 13. Average number of orphans created per link by the *LO* algorithm for the Mesh-torus network.

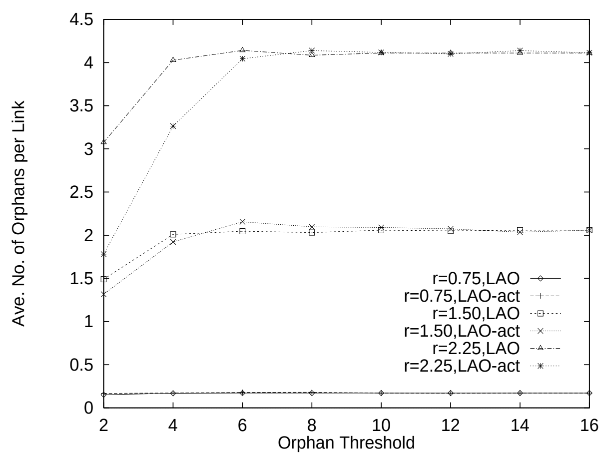

Fig. 12. Average number of orphans created per link by the *LAO* and *LAO-actual* algorithms for the ARPA-2 network.

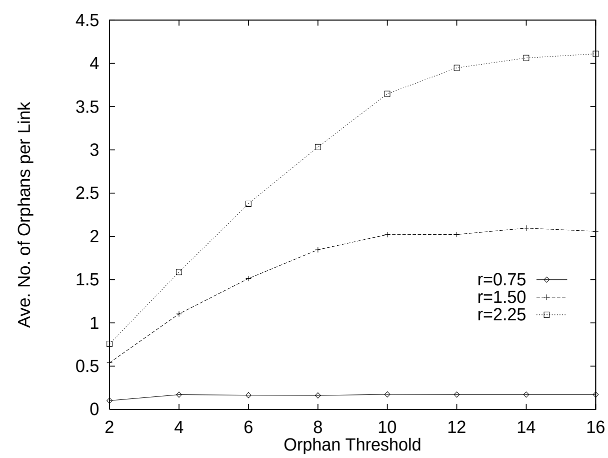

Fig. 14. Average number of orphans created per link by the *LO* algorithm for the ARPA-2 network.

simulation experiments on the Mesh-torus and ARPA-2 networks. The results show that under light load conditions, more than 90% performance gain can be achieved at the expense of less than 10% reduction in restoration guarantee. Our results also show that even under moderate and heavy load conditions, the performance gain is more when compared to the reduction in failure restoration guarantee.

References

[1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WANs", *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171-1182, July 1992.

[2] R. Ramaswami and K.N. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks", *IEEE/ACM Transactions on Networking*, vol. 3. no. 5, pp. 489-500, October 1995.

[3] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength Routed Optical Networks", *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 903-908, June 1996.

[4] A. Mokhtar and M. Azizoglu, "Adaptive Wavelength Routing in All-Optical Networks", *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 197-206, April 1998.

[5] S. Subramaniam, M. Azizoglu, and A.K. Somani, "All-Optical Networks With Sparse Wavelength Conversion", *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 544-557, August 1996.

[6] H. Harai, M. Murata, and H. Miyahara, "Performance of Alternate Routing Methods in All-Optical Switching Networks", In Proc. of *INFOCOM'97*, 1997.

[7] E. Karasan and E. Ayanoglu, "Performance of WDM Transport Networks", *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1081-1096, September 1998.

[8] M. Veeraraghavan, A. Rodriguez-Moral, and J. Anderson, "Integrated IP/WDM routing", In Proc. of *Network Management Symposium'98*, May, 1998.

[9] R. Ramaswami and A. Segall, "Distributed Network Control for Wavelength Routed Optical Networks", In Proc. of *INFOCOM'96*, 1996.

[10] S. Han and K.G. Shin, "Efficient Spare Resource Allocation for Fast Restoration of Real-Time Channels from Network Component Failures", In Proc. of *Real-Time Systems Symposium, RTSS*, 1997.

[11] C.S. Li and R. Ramaswami, "Automatic Fault Detection, Isolation, and Recovery in Transparent All-Optical Networks", *IEEE/OSA Journal of Lightwave Technology*, vol. 15, no. 10, pp. 1784-1793, October 1997.

[12] N. Nagatsu, S. Okamoto, and K. Sato, "Optical Path Cross-Connect System Scale Evaluation Using Path Accommodation Design for Restricted Wavelength Multiplexing", *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 893-902, June 1996.

[13] M. Alanyali and E. Ayanoglu, "Provisioning Algorithms for WDM Optical Networks", In Proc. of *INFOCOM'98*, pp. 910-918, 1998.

[14] S. Bandyopadhyay, A. Sengupta, and A. Jaekel, "Fault-tolerant Routing Scheme for All-Optical Networks", In Proc. of *SPIE Conference on All-Optical Communication Systems*, 1998.