Backup Reprovisioning to Remedy the Effect of Multiple Link Failures in WDM Mesh Networks

Jing Zhang, *Student Member, IEEE*, Keyao Zhu, *Member, IEEE*,
and Biswanath Mukherjee, *Senior Member, IEEE*

***Abstract*— As networks grow in size and complexity, both the probability and the impact of failures increase. The pre-allocated backup bandwidth, which has been widely investigated in the literature, may not be able to provide full protection guarantee when multiple failures occur in a network. In this study, we consider multiple concurrent failures where concurrent means that a new failure occurs before a previous failure is repaired. To combat the effect of multiple concurrent failures, new backups can be reprovisioned after one failure such that the next potential failure can be handled effectively and efficiently.**

We consider dynamic traffic where a pair of link-disjoint primary and backup paths is provisioned when a new connection request arrives. After a failure occurs, the affected connections switch traffic from their primary paths to backup paths. To protect against next potential failure, we reprovision new backups for connections that become unprotected or vulnerable because of losing their primary or their backup due to the previous failure or due to backup resource sharing. This approach is called Minimal Backup Reprovisioning (MBR). An alternative approach is to globally rearrange backups for all connections after one failure occurs, which is called Global Backup Reprovisioning (GBR).

Backup reprovisioning can be performed whenever the network's *state* changes, e.g., (1) when a new request arrives, (2) when an existing connection terminates, (3) when a network failure occurs, (4) when a failed link/node is repaired, etc., to utilize the available resources more efficiently or to recover quickly from the next failure. In this study, we perform MBR or GBR after one network failure occurs to protect against the next potential failure in a wavelength-convertible WDM mesh network. The link-vector network model which can maximally explore the backup-sharing potential is assumed in this study. We then analyze the complexity of MBR and GBR under such a network model. A reprovisioning algorithm is proposed for MBR which can significantly reduce the connection vulnerability without the knowledge of the location of the next failure. In GBR, both integer linear program (ILP) and heuristic-based approaches are proposed. We compare capacity requirement and computational complexity of MBR to that of GBR through numerical examples. MBR demonstrates a good tradeoff between complexity and capacity efficiency to handle multiple concurrent failures.

***Index Terms*— Optical network, WDM, protection, backup reprovisioning, multiple link failures, backup rearrangement.**

Manuscript received November 22, 2004; revised September 20, 2005. This work was supported in part by the National Science Foundation (NSF) under Grant No. ANI-9805285. A short, summarized version of this paper was presented at the IEEE International Conference on Communications (ICC), Paris, France, in June 2004.

Jing Zhang is with Sun Microsystems, Menlo Park, CA 94025 USA (e-mail: j.zhang@sun.com).

Keyao Zhu is with Brion Tech. Inc., Santa Clara, CA 95054 USA (e-mail: kzhu@briontech.com).

Biswanath Mukherjee is with the Computer Science Department, University of California, Davis, CA 95616 USA (e-mail: mukherje@cs.ucdavis.edu).

Digital Object Identifier 10.1109/JSAC-OCN.2006.19404.

I. Introduction

IN AN OPTICAL network employing wavelength-division multiplexing (WDM), the failure of a single fiber link may lead to tremendous data loss since a single fiber link can carry a huge amount of data (on the order of terabits per second). Therefore, network survivability is an important problem in network design and its real-time operation. In order to reduce the data loss, various protection and restoration mechanisms have been proposed and studied in the literature to recover traffic after a failure occurs and before the failure is physically repaired (see, e.g., [1], [2], [3], [4], [5]).

In a *protection* scheme, extra bandwidth is reserved when the connection is provisioned. Usually, a pair of paths is provided to a connection: one is used to carry traffic during normal operation, referred to as the *primary path*, and the other path, referred to as the *backup path*, is reserved and will be activated after a failure occurs on the primary path. The primary and backup paths are usually link or node disjoint, which guarantees that at least one path is available when any single link or node (except the two end nodes) fails in the network. The backup resources are usually provisioned and reserved when primary resources are provisioned and established, and torn down when the connection leaves the network.

However, the network's *state* changes when a new event occur, e.g., (1) when a new request arrives, (2) when an existing connection terminates, (3) when a network failure occurs, (4) when a failed link/node is repaired, etc. Resource and link availabilities are updated at these instances. The initially-provisioned backup resources may not be optimal under the new (changed) network state. To adapt to the changing network state, backup resources can be reprovisioned periodically, when a path is blocked, when resource utilization exceeds a threshold, or when the network state changes to achieve better resource utilization.

As networks grow in size and complexity, both the probability and the impact of failures increase. The pre-allocated backup bandwidth may not be able to provide protection guarantee when multiple failures occur in a network. In this study, we allow the possibility of multiple concurrent failures, where concurrent means that a new failure may occur before a previous failure has been repaired. We mainly focus on multiple link failures since link failure is the dominant failure in optical networks. Considering two concurrent failures, the one which occurs earlier is referred to as the old failure and the one which occurs after the old failure is called the new failure.

Then, there exist the following three scenarios under which a disrupted connection A suffers traffic loss (if no dynamic restoration procedure is triggered) after the new failure occurs:

1) The old failure affects the primary path of A and the new failure affects the backup path of A.
2) The old failure affects the backup path of A and the new failure affects the primary path of A.
3) (Assume that connection A shares backup bandwidth with connection B.) The old failure affects the primary path of B and the new failure affects the primary path of A. In this case, the shared backup bandwidth has been taken by B when the primary path of A fails.

To combat the effect of multiple concurrent failures, we can reprovision new backups for connections that become unprotected or vulnerable because of losing their primary or their backup due to the previous failure or due to backup resource sharing, so that connections can recover quickly from the next potential failure. This approach is called Minimal Backup Reprovisioning (MBR). An alternative approach is to globally rearrange backups for all connections after one failure occurs, which is called Global Backup Reprovisioning (GBR). Backup reprovisioning can be performed whenever the network's state changes to utilize resource more efficiently or to recover quickly from the next failure. In this study, we perform MBR or GBR after a network failure occurs to protect against the next potential failure, and we compare the characteristics of the two approaches.

With backup reprovisioning, it is not necessary to constrain or predict the number of concurrent failures because fast protection switching can be triggered as long as the backup reprovisioning succeeds before the next failure occurs. In addition, the work in [6] has showed that the chance of a second failure impacting the network within the Mean Time to Repair (MTTR) of the first failure is proportional to the MTTR and is about 4 in 100 when MTTR is 12 hours, which demonstrates that reprovisioning is a beneficial technique to consider.

The remainder of the paper is organized as follows. Section II discusses the prior work in the literature to handle multiple-failure scenarios. Section III discusses the network model and our contributions. Section IV-A explains the evaluation measures. We propose a new backup-reprovisioning algorithm for MBR in Section IV-B. In Section V, we propose both integer linear program (ILP) and heuristic-based approaches for GBR. In Section VI, we show the results from evaluation measures, demonstrate the effectiveness of our MBR approach, and compare the capacity requirement and the computational complexity of MBR to that of GBR using illustrative numerical examples. Finally, Section VII concludes this study.

II. Prior Work

A variety of schemes have been studied in the literature to handle multiple-failure scenarios, especially for double-link failures[1] [4], [6], [7], [8], [9], [10], [11].

[1]Double-link failure is referred to as the case of two independent link failures where the second failure occurs before the first failure is repaired.

In [7], new protection techniques are designed such that they can tolerate double-link failures. The authors assume that any two arbitrary links may fail in any order. One or two backup paths are provided to each link, and optimization techniques are proposed to design backup paths such that the capacity can be minimized and 100% recovery from double-link failures using backup paths can be achieved. The work in [4] evaluates the restorability of span-restorable mesh networks when double-link failures occur. One connection can be restored on the fly if both its primary path and precomputed backup path get affected when double-link failures occur. The restorability of a network is defined as the average fraction of failed working capacity that can be restored within the spare capacity. It is reported that single-failure-designed mesh networks inherently have high levels of double-link-failure restorability.

The work in [8] evaluates online protection reconfiguration where reconfiguration is performed dynamically after a failure. The results show that dynamic reconfiguration can be implemented with little additional capacity compared to the network without reconfiguration. The authors in [9] analyze the restorability and capacity consumption of path protection and rerouting mechanisms in a network planned for double-link failures. The rerouting scheme using stub-release achieves high restorability. The work in [10] quantitatively measures the degree to which a network can recover from double-link failures using link-restoration algorithms. The work in [11] explores the tradeoff between capacity and robustness to double-link failures using link-restoration algorithms. The authors in [6] consider the benefits and operational complexity of preemptive reprovisioning where a backup path is reprovisioned in advance of a second failure to reduce the time to recover services from the second failure.

In general, instead of dynamically restoring traffic after the second failure, an alternative approach can be referred to as Reprovisioning/Reconfiguration Before the Second Failure (RBSF) [6], [8], [9]. In RBSF, the first failure is handled using predesigned protection bandwidth so the recovery from the first failure is guaranteed, and, then, new backup paths are reprovisioned before the second failure using spare or additional capacity in the network for the connections which may be unrecoverable using predesigned protection bandwidth (e.g., connection A illustrated in Section I). The affected connections will directly take the reprovisioned backup routes as soon as the second failure occurs, which speeds up the traffic-recovery procedure compared to dynamic restoration.

However, the reprovisioning in RBSF may be quite complex since the second failure could be on any link in the network except the one already failed. For example, in Fig. 1, the primary paths of connections A_1, A_2, and A_3 traverse link l_0, and the backup path of connection A_i traverses link l_i, $1 \leq i \leq 3$. Let us assume that link l_0 fails. Then, all three connections switch traffic to their backup. Thus, depending on which link the second failure breaks (l_1, l_2, or l_3), only one connection needs to be dynamically restored in this example if restoration is applied, while if using RBSF, all three connections need to be reprovisioned for new backups since the next failure could be on any link. In addition, the connections that share backup wavelengths with A_1, A_2, or A_3 also need to be

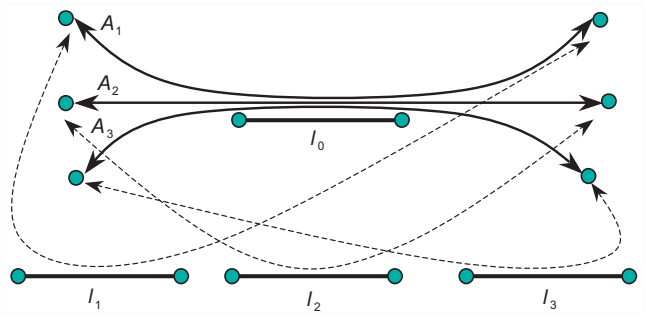
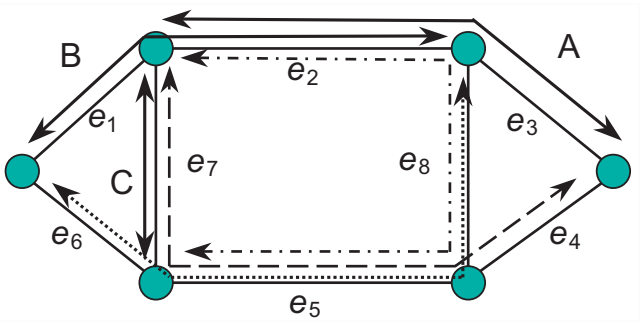


Fig. 1. An example of the complexity of RBSF, compared to dynamic restoration.



(a) Sample network and connections.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	$\nu^*_{e_5}$	$\overline{\nu}_{e_5}$
Before failure: e_5	(1,	2,	1,	0,	0,	0,	1,	0,	2)	–
Case I (after e_2 fails): e_5	(0,	0,	0,	0,	0,	0,	1,	0,	1)	0
Case II (after e_7 fails): e_5	(1,	2,	1,	0,	0,	0,	0,	0,	2)	1

(b) Link vector for e_5, before and after failure.

Fig. 2. Example of backup wavelength contention, shown using link vector.

reprovisioned (which are not shown in Fig. 1). Comparing RBSF with dynamic restoration, the tradeoff is complexity versus speed.

In this paper, we apply the idea of RBSF to multiple link failures, i.e., we do not constrain or predict the number of concurrent failures. New backups are reprovisioned after each failure such that a large number of connections (if not all) have backups when the next failure occurs. We provide a comprehensive study for both Minimal Backup Reprovisioning (MBR) and Global Backup Reprovisioning (GBR) in a wavelength-convertible WDM mesh network. The pros and cons of MBR and GBR are discussed. The link-vector network model which can maximally explore the backup-sharing potential is assumed in this study. We then analyze the complexity of both approaches under such a network model and propose effective reprovisioning algorithms.

III. Problem Statement and Our Contributions

The goal of this work is to study the feasibility, scalability, and capacity requirement for conducting backup reprovisioning of vulnerable connections (in MBR) or all connections (in GBR) against the next potential failure after one failure occurs. We assume a shared-path protection scheme since it is shown to be capacity efficient in mesh networks [5]. When a connection request arrives to the network, the shortest path is used as the primary path, and the backup path is routed such that it is link-disjoint to the primary path and shares backup capacity as much as possible with existing connections. When a link goes "down", all the connections traversing the link are disrupted. The source node of a failed connection signals the intermediate nodes along its backup path and sends traffic to the backup path after the intermediate nodes configure the switches. When protection switching is performed for all failed connections, the network is in a "temporarily stable" (TS) state in the sense that all connections are currently "up" but some of them are vulnerable to the next failure. Thus, we need to identify the connections which are vulnerable for the next possible failure and perform backup reprovisioning for them.

The link-vector scheme has been widely applied in various studies (e.g., the conflict vector in [3], the aggregated square matrix in [12], the "bucket" link metric in [13], and the conflict set in [14]) to identify the sharing potential between backup paths, especially in a wavelength-convertible network. Essentially, the idea is to associate a vector with each link in the network, identifying the number of backup wavelengths to be reserved on this link to protect against failures of other links. The link vector ν_e for link e can be represented as an integer set, $\{\nu_e^{e'} \mid \forall e' \in E, 0 \leq \nu_e^{e'} \leq \lambda(e')\}$, where E is the set of links; $\lambda(e')$ specifies the number of wavelengths on link e'; and $\nu_e^{e'}$ specifies the number of working lightpaths that traverse link e' and are protected by link e (i.e., their corresponding backup lightpaths traverse link e).

Through such a simple data structure, the link vector captures the necessary information on the sharing potential offered by each link. The number of wavelengths which need to be reserved for backup lightpaths on link e is thus $\nu_e^* = \max_{\forall e'}\{\nu_e^{e'}\}$. Therefore, using the link vector, we can simply reserve ν_e^* wavelengths on link e as backup wavelengths. However, when a failure occurs and connections switch traffic to backup, some backup wavelengths on e may be activated. After updating ν_e^* according to the current network state, one may find that $\overline{\nu}_e \leq \nu_e^*$, where $\overline{\nu}_e$ is the number of reserved backup wavelengths still available on link e after protection switching. Figure 2(a) shows such an example where three connections (A, B, and C) are in the network. (All connections are assumed to be bidirectional.) Solid lines show the primary paths and dashed lines show the backup paths. Figure 2(b) shows the link vector for e_5 according to current network state: e_5 reserves two backup wavelengths in order to protect connections A, B, and C. Consider the two cases below.

- Case I: If link e_2 fails first, two backup wavelengths are activated for connections A and B. The new vector for e_5 is given in Fig. 2(b), Case I. We find that $\nu^*_{e_5} = 1$ but no reserved backup wavelength is available after e_2 fails, i.e., $\overline{\nu}_{e_5} = 0$. If e_7 fails next, connection C cannot be recovered.
- Case II: If link e_7 fails first, one backup wavelength is activated for connection C. The new vector for link e_5 is given in Fig. 2(b), Case II. We find that $\nu^*_{e_5} = 2$ but $\overline{\nu}_{e_5} = 1$. If e_2 fails next, connections A and B will contend for the only backup wavelength on e_5.

To resolve the contention, one simple solution is to reserve more backup wavelengths on contending links, e.g., link e_5 in Fig. 2. Unfortunately, the reservation may fail due to limited bandwidth on links, especially when the network is heavily loaded. Our simulation results demonstrate that 15-20% of contending links do not have enough free capacity to resolve the contention when the average link load is 70-80%. Even if all the contending links can successfully resolve the contentions by reserving new backup wavelengths, it is not a cost-effective solution as some links may use up the free capacity which leads to high blocking of new incoming connections.

In this study, we propose to first investigate minimal backup reprovisioning (MBR) to resolve the contention of backup wavelengths for the next possible failure. We evaluate the fraction of connections which are involved in the contention, and which are defined as *vulnerable connections*. Then, some of the vulnerable connections are picked and their backups are reprovisioned. We investigate approaches on how to pick fewer number of vulnerable connections to reduce contentions. We first assume that no new capacity can be added to the network for reprovisioning purpose. The new provisioned backups can share bandwidth with existing backups of other connections, or utilize the spare bandwidth (if available) on links if sharing is not possible. However, due to limited link capacity, the backup reprovisioning may fail. We then study the reprovisioning success rate (denoted as RSR) using different policies under different network load. Obviously, a network needs to be at least 2-edge-connected[2] in order for a pair of link-disjoint paths to exist between any node pair. In what follows, we assume that the topology is still 2-edge-connected after one failure.

In order to successfully reprovision all the vulnerable connections, additional capacity may be needed. In this study, we also investigate the capacity requirement for MBR with 100% RSR. Another possibility to resolve the backup contention is to perform a backup rearrangement where the backups for all the existing connections are globally recomputed such that the backup contention is eliminated, and at the same time, the usage of the backup capacity is minimized. Both ILP and heuristic-based approaches are studied for the global backup reprovisioning (GBR). We compare the capacity requirement and the computational complexity of the MBR approach with 100% RSR to that of the GBR approach through numerical examples.

Note that we don't need to specify the location of the next failure using our reprovisioning approach. On the contrary, all possible failures are taken care of through reprovisioning. The network control and management system can activate the reprovisioning procedure after one failure occurs. The new backup routes could be stored in a centralized database and retrieved to restore traffic when the next failure occurs. As we have discussed above, backup reprovisioning is scalable in terms of the number of concurrent failures since we can easily conduct backup reprovisioning after each failure. Furthermore, the reprovisioning can be pre-emptive in the sense that the reprovisioned backup paths can be discarded when the first failure gets fixed and the traffic streams are reverted back to their primary paths. However, a network operator may not always choose to revert traffic from backup to primary after a failure is repaired since one additional "hit" may affect the customers' data flows. So, backup reprovisioning may be preferred for a network employing a shared protection scheme, and the new backup paths are used to protect against next failure.

Compared to [6], we consider backup reprovisioning in a network where the link vector is applied for shared-path protection. Using link vector, the network is modeled in a more general and efficient way and the sharing potential between backup paths can be maximally explored (while the work in [6] does not assume such a network model). However, using link vector also adds complexity to reprovisioning because more connections may become vulnerable and may need to be reprovisioned. In this study, we propose intelligent selection policies to reduce the number of connections that need to be reprovisioned, and we investigate their performance.

IV. Minimal Backup Reprovisioning (MBR)

A. Evaluation Measure

As we mentioned before, when protection switching is performed for all failed connections, the network is in a "temporarily stable" (TS) state since all connections are currently "up" but some of them are vulnerable to the next failure. A connection can be classified into one of three groups when the network is in TS state:

1) *Unprotected connection*: Connection that lost its primary or backup in the prior (or first) failure so it is unprotected with respect to the next failure.
2) *Vulnerable connection*: Connection that wasn't affected by the prior failure (both its primary and backup), but, on some link e on its backup path, we have $\overline{\nu}_e < \nu_e^*$. The backup contention makes the connection vulnerable to the next failure.
3) *Unaffected connection*: Connection that wasn't affected by the prior failure, and on all the links which its backup traverses, we have $\overline{\nu}_e \geq \nu_e^*$.

Note that all unprotected connections need to be reprovisioned. However, not all vulnerable connections need to be reprovisioned. For example, in Fig. 2, Case II, reprovisioning a backup for either A or B can resolve the contention on link e_5. We first perform a quantitative measurement to evaluate the fraction of connections in each group, which could help us to understand the complexity of the problem and place an upper bound for the reprovisioning workload. We define *connection vulnerability* as the ratio between the number of vulnerable connections to the total number of connections, which gives the probability that a connection is vulnerable to the next failure.

Let us use the following notations: the set of connections currently carried by the network is T ($|T|$ is the size of T), the average hop distance (in terms of number of links traversed) for primary path of a connection is P, the average hop distance for backup path is B, and the network has L links. So, the fraction of unprotected connections when link e fails, denoted

[2]A graph is *k-edge-connected* if any subgraph formed by removing any $k-1$ edges is still connected.

by U_e, is the fraction of connections whose primary paths or backup paths traverse e and can be represented as follows (where $\frac{|T|\times P}{L}$ gives the average number of connections whose primary paths traverse e, and $\frac{|T|\times B}{L}$ gives the average number of connections whose backup paths traverse e):

$$U_e = \frac{\frac{|T|\times P}{L} + \frac{|T|\times B}{L}}{|T|} = \frac{P+B}{L} \quad (1)$$

We find that U_e is determined by P, B, and L.

We define *shareability of a backup wavelength* to be the number of connections that are sharing that backup wavelength. If the link-vector model is applied, shareability of a backup wavelength on link e is defined as the number of connections (denoted as N_e) whose backups are traversing link e divided by the number of backup wavelengths reserved on e. So, all the backup wavelengths on the same link have equal shareability.

Shareability of a network is the average value of the shareability of each backup wavelength, which indicates the number of connections, on average, which share the same backup wavelength in the network. If this value is large, more backup paths are packed together, which implies that backup bandwidth is utilized efficiently. However, a large number of connections become vulnerable after one failure if backup paths are packed too tight. Thus, we should bound the shareability of each wavelength to a reasonable value.

Hence, we introduce a parameter called *Maximal Allowed Shareability* (MAS) to place an upper bound on the shareability of each backup wavelength. So, the number of backup wavelengths that need to be reserved on link e (i.e., ν_e^*) should be constrained by $\nu_e^* \geq N_e/MAS$. Thus, ν_e^* is computed according to the updated link vector and MAS whenever a new connection is provisioned in the network. In our evaluation measure, we study the effect of MAS on connection vulnerability, and design a proper MAS for the network, which represents a tradeoff between bandwidth efficiency and connection vulnerability.

B. Reprovisioning Approach

We are given a network, the primary and backup paths of connections currently carried by the network, and the failed link l_f; so the group of vulnerable connections (due to the failure of link l_f) can be identified. The problem is to select the vulnerable connections to reprovision their backups for reducing the connection vulnerability with respect to the next failure. This problem can be challenging, as our primary objective is to reduce the connection vulnerability as much as possible. In addition, the number of vulnerable connections could be large, e.g., after the prior failure, around 40% of connections may be vulnerable when the average network link load is around 50%, averaged over all possible link failures, according to the numerical examples in our simulation experiments. It would be beneficial to reduce the workload for reprovisioning if we can select a fewer number of connections to reprovision while still reducing the connection vulnerability as much as possible.

We first list the notations used in this study (some of which have been defined and explained before but are repeated here for completeness).

- e^*: the link on which the prior failure occurs. We use e to represent any other link besides e^*.
- N_e: number of connections whose backups are traversing link e.
- $N_e^{e^*}$: number of connections whose primary uses e^* and backup uses e.
- ν_e^*: number of backup wavelengths that need to be reserved on e, computed according to link vector, N_e, and MAS.
- $\overline{\nu}_e$: number of reserved backup wavelengths still available on e after e^* fails.
- $f(e)$: number of free wavelengths on e.

We propose and study three policies for selecting vulnerable connections to reprovision:

- *Random*: Randomly pick one connection from the vulnerable group.
- *Longest Backup (LB)*: Pick a vulnerable connection with the longest backup path.
- *Most Violations (MV)*: Count the number of violations of a connection as the number of backup links that have contention (i.e., $\overline{\nu}_e \leq \nu_e^*$). Pick a connection with the largest number of violations.

If reprovisioning one connection can resolve the contention on multiple links, fewer connections may need to be reprovisioned. We hope to achieve this by choosing the connections with the longest backup or the most violations.

Assume that link e^* fails, so all the connections whose primaries use e^* are disturbed. We switch these connections to their backups, and we free up the primary wavelength on each link (except e^*) along their original primary paths. Then, we update the vector of each link accordingly. Algorithm 1 describes how to calculate the number of remaining backup wavelengths on e (i.e., $\overline{\nu}_e, \forall e \in E$), and how to reprovision connections. Equation (2) defines the cost function when computing a new backup route for the selected connection using a shortest-path algorithm. The cost function ensures that a) the backup path is link-disjoint to the primary path and the failed link cannot be used; and b) if an existing backup wavelength on a link can be shared by this connection, the link has cost ϵ (where ϵ is a small positive number and it is used to avoid loops and unnecessarily long paths when a shortest path is computed using the cost function), 1, or ∞ otherwise (link cost is 1 if there is at least one free wavelength on this link and ∞ if not).

The reprovisioning algorithm is composed of (at most) $|T|$ runs of a standard shortest-path algorithm, where T is the set of connections currently carried by the network. Thus, the time for reprovisioning should be far less than MTTR of the prior failure. Note that Algorithm 1 may not succeed when recomputing the backup for an unprotected connection or a selected vulnerable connection due to limited bandwidth on each link. When one vulnerable connection is successfully reprovisioned, we hope that the size of the vulnerable group can be reduced by more than one. Even if a vulnerable connection A cannot be successfully reprovisioned, it may become "invulnerable" at the end of reprovisioning. This is because other vulnerable connections that contend for backup wavelengths with A on some links may be successfully reprovisioned later on.

Algorithm 1 Reprovisioning Algorithm

1) $\forall e \in E, \overline{\nu}_e = \nu_e^* - N_e^{e^*}$.
2) Identify unprotected connections and vulnerable connections.
3) Reprovision a new backup for each unprotected connection. When computing the backup route for connection A, define the cost of link e, $\forall e \in E$, as follows (assume that primary path of A traverses the links $e_1, e_2, ..., e_m$):

$$Cost(e) = \begin{cases} \infty & \text{if } e \text{ is on primary path of A or } e = e^*, \\ \epsilon & \text{if } \nu_e^{e_1} < \overline{\nu}_e, ..., \nu_e^{e_m} < \overline{\nu}_e, \\ & \text{and } (N_e + 1)/\overline{\nu}_e \le MAS, \\ 1 & \text{if } f(e) > 0, \\ \infty & \text{otherwise.} \end{cases} \quad (2)$$

   Apply a shortest-path algorithm to compute the backup route using the new link costs. Update link vector and ν_e^* accordingly if a new backup can be found and allocate a new backup wavelength if $\nu_e^* > \overline{\nu}_e$ or $N_e/\overline{\nu}_e > MAS$, $\forall e \in E$.
4) Select one vulnerable connection according to one of the policies described above (Random, LB, or MV). Compute a new backup route for the selected connection using the shortest-path algorithm where the cost of each link is defined by Eqn. (2).
5) Recompute the vulnerable group but exclude the connections that have been reprovisioned whether or not the reprovisioning succeeded; go to Step 4 until the size of the vulnerable group becomes 0.

V. Global Backup Reprovisioning (GBR)

The idea of backup rearrangement (also called backup relocation, migration, or adjustment) has been applied in various studies for various purposes. For example, in [15], when backup capacity utilization exceeds a preset threshold, backup paths are reassigned from a precomputed path set to optimize the usage of backup capacity. Reference [16] presents a backup relocation policy to migrate backup connections onto new routes to accommodate requests that would otherwise have been rejected due to limited usage of wavelength converters. In [17], a backup-path migration scheme has been proposed where backup paths are migrated to paths selected from a set of k precomputed paths.

In this study, we perform a global backup reprovisioning (GBR) for all connections after one failure occurs to resolve the backup contention and to optimize the usage of backup capacity. We formulate the problem into an integer linear program (ILP). The ILP formulation is summarized below.

- *Given:*
  - $G = (V, E)$: network topology with node set V and edge set E. The failed link has been removed from E.
  - $T = \{t = \langle s, d \rangle\}$, the set of connections currently in the network where s is the source, and d is the destination. Assume that each t requires one full wavelength capacity.
  - MAS: maximal allowed shareability in the network.
  - F_{ij}: number of free wavelengths on link (i, j).
  - R_{ij}^{tp}: route of the primary path of connection t: $R_{ij}^{tp} = 1$ if the primary path of connection t is routed through link (i, j); otherwise, $R_{ij}^{tp} = 0$. Note that the unprotected connections have switched traffic to their backups so their primary paths are taking the backup routes now.
- *Variables:*
  - X_{ij}^{tb}: $X_{ij}^{tb} = 1$ if the backup path of connection t is routed through link (i, j); otherwise, $X_{ij}^{tb} = 0$.
  - K_{ij}: number of backup wavelengths that need to be reserved on link (i, j), $K_{ij} \ge 0$.
- *Objective:* Minimize the total wavelength links used by backup paths:

$$Minimize: \sum_{(i,j)\in E} K_{ij} + \alpha \sum_{t\in T} \sum_{(i,j)\in E} X_{ij}^{tb} \quad (3)$$

- *Constraints:*
  - On backup route flow-conservation constraints:

$$\sum_{(k,j)\in E} X_{kj}^{tb} - \sum_{(i,k)\in E} X_{ik}^{tb} = 0 \quad \forall k \in V, t \in T, k \ne s, d \quad (4)$$

$$\sum_{(s,j)\in E} X_{sj}^{tb} - \sum_{(i,s)\in E} X_{is}^{tb} = 1 \quad \forall t \in T \quad (5)$$

  - On primary and backup link-disjoint constraints:

$$R_{ij}^{tp} + X_{ij}^{tb} \le 1 \quad \forall (i, j) \in E, t \in T \quad (6)$$

  - On network shareability constraints:

$$\frac{1}{MAS} \times \sum_{t\in T} X_{ij}^{tb} \le K_{ij} \quad \forall (i, j) \in E \quad (7)$$

  - On link capacity constraints:

$$K_{ij} \le F_{ij} \quad \forall (i, j) \in E \quad (8)$$

  - Computation of K_{ij}:

$$\sum_{t\in T} (R_{i_1 j_1}^{tp} \times X_{ij}^{tb}) \le K_{ij} \quad \forall (i, j), (i_1, j_1) \in E, (i, j) \ne (i_1, j_1) \quad (9)$$

Note that the second term $\alpha \sum_{t\in T} \sum_{(i,j)\in E} X_{ij}^{tb}$ in Eqn. (3) tries to avoid loops and unnecessarily long paths in routing, and α is a positive number which is assigned a small value such that minimizing $\sum_{(i,j)\in E} K_{ij}$ (i.e., total capacity allocated to backup paths) is of higher priority. Also, in the backup route flow-conservation constraints, a constraint on the sink is superfluous, and therefore not included so as to reduce the problem size.

In Eqn. (9), $R_{i_1 j_1}^{tp}$ is a constant (either 0 or 1), so the equation is linear. $R_{i_1 j_1}^{tp} \times X_{ij}^{tb}$ is 1 only when both $R_{i_1 j_1}^{tp}$ and X_{ij}^{tb} are 1, otherwise it has value 0. Through Eqn. (9), we constrain that the number of connections whose primary paths traverse link (i_1, j_1) and whose backup paths traverse link (i, j) should be less than or equal to the number of backup wavelengths reserved on link (i, j) (i.e., K_{ij}), $\forall (i, j), (i_1, j_1) \in E$.

Using the ILP, we minimize the total capacity allocated to the backup paths. Note that the backup capacity optimized by

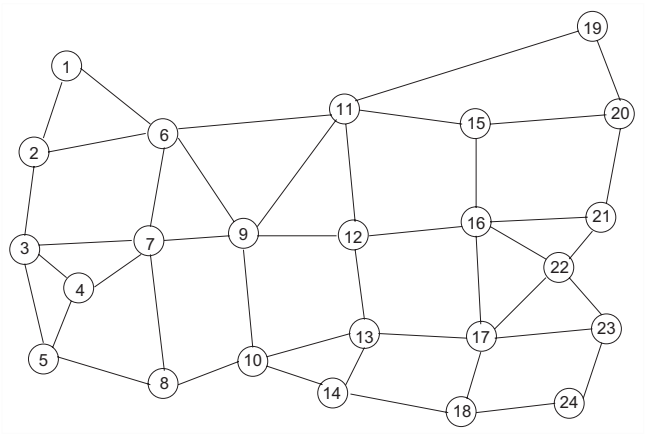


Fig. 3. A sample network topology.

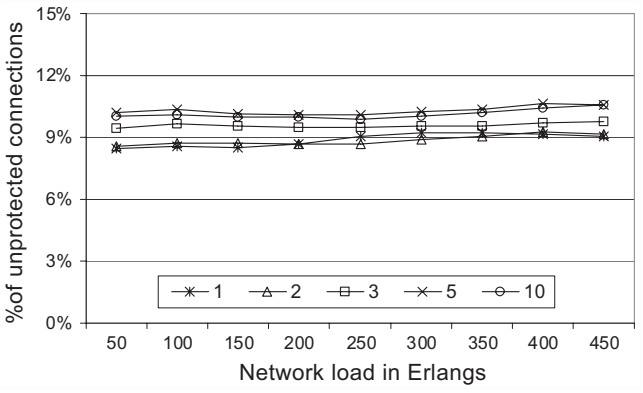


Fig. 4. Percentage of unprotected connections for MBR with MAS = 1, 2, 3, 5, 10.

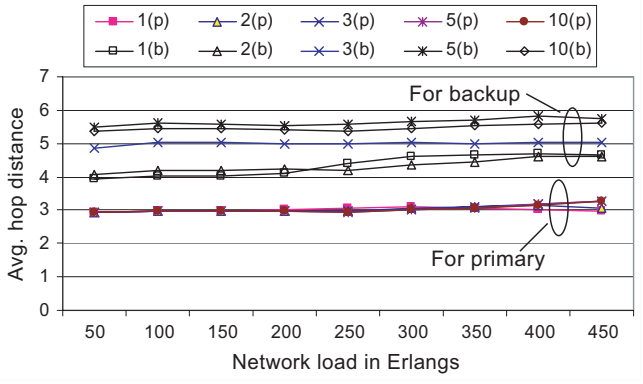


Fig. 5. Average primary and backup hop distance for MBR with MAS = 1, 2, 3, 5, 10.

the ILP may not always be less than the total backup capacity provisioned before the failure as the failed link cannot be used in the optimization. Given the time complexity of the ILP approach, we also propose a heuristic algorithm to perform backup rearrangement. In our heuristic, Eqn. (2) is used as the cost function when computing a new backup route for a selected connection using a shortest-path algorithm, and we randomly generate S different sequences for connections to compute backups. The solution with the minimum backup capacity consumption is selected.

Note that $|T|$ is the number of connections currently carried by the network, N is the number of nodes, and L is the number of links in the network. The ILP presented in this section has $L + |T| \times L$ variables and $|T| \times (N + L - 1) + L \times (L + 1)$ constraints while the complexity of the heuristic algorithm is $S \times |T| \times (L + N \log N)$ and the complexity of MBR is $|T| \times (L + N \log N)$, given that the complexity of the shortest-path algorithm is $(L + N \log N)$.

VI. Illustrative Numerical Examples

In this section, we present some numerical examples to illustrate the performance of our reprovisioning algorithms and to compare the capacity requirement and the computational complexity of the MBR approach with 100% RSR to that of the GBR approach. Figure 3 shows one of the network topologies we used in this study, which is a representative US nationwide network with 24 nodes and 43 bidirectional links. Each link is assumed to have 32 wavelength channels in each direction. Connection arrivals are Poisson and they are uniformly distributed among all source-destination pairs. The holding time of each connection follows a negative exponential distribution. In addition, we assume that the network has full wavelength-conversion capability.

A. Results from Minimal Backup Reprovisioning (MBR)

1) Evaluation Measure: We simulate the failure of one unidirectional link, and then we calculate the percentage of unprotected connections and connection vulnerability after the failed connections switch their traffic to backups. We vary MAS over five different values – 1, 2, 3, 5, 10 – to study its effect on connection vulnerability. Please note that $MAS = 1$ is equavalent to dedicated-path protection as no sharing is allowed. The results are averaged over all possible link failures except the links after removing which the topology is not 2-edge-connected.

In Fig. 4, we compare the percentage of unprotected connections (denoted as $\%U$) for MBR for different values of MAS, i.e., $MAS = 1$, 2, 3, 5, or 10. We observe that $\%U$ increases when MAS increases but varies slightly when network load increases. According to Eqn. (1), $\%U$ is mainly determined by the average hop distance of primary (P) and backup (B) paths, which are shown in Fig. 5. We observe that P does not change much as load and MAS change, but B shows similar trends as $\%U$, i.e., it increases when MAS increases but varies slightly when network load increases.

Figure 6 shows the connection vulnerability for MBR before reprovisioning. We observe that the connection vulnerability increases when load increases. This is because the sharing potential can be explored more when load increases. For example, two wavelengths need to be reserved on link e_5 (as $\nu^*_{e_5} = 2$) according to the link vector shown in Fig. 2(b) but only link e_2 requires two wavelengths on e_5 to protect it (as only $\nu^{e_2}_{e_5} = 2$). If all the links (except e_5) require two backup wavelengths on e_5 (i.e., $\nu^{e'}_{e_5} = 2, \forall e' \in E, e' \neq e_5$), the sharing potential on e_5 is fully utilized. We can approach such an ideal case when load is high since routes are more diverse when more connections are in the network. We also notice that more connections are vulnerable when MAS is

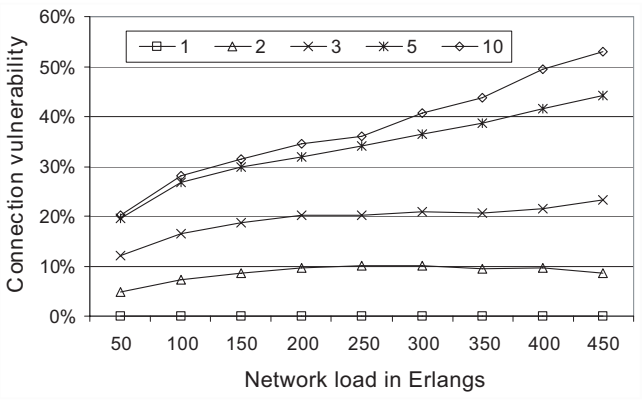


Fig. 6. Connection vulnerability before reprovisioning for MBR with MAS = 1, 2, 3, 5, 10.

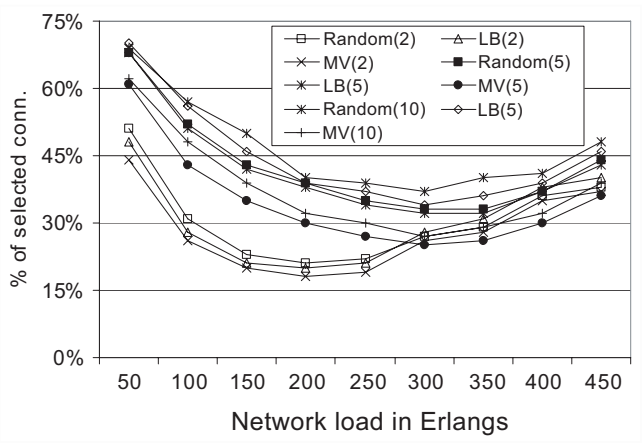


Fig. 7. Percentage of selected vulnerable connections for MBR under different policies for MAS=2, 5, 10.

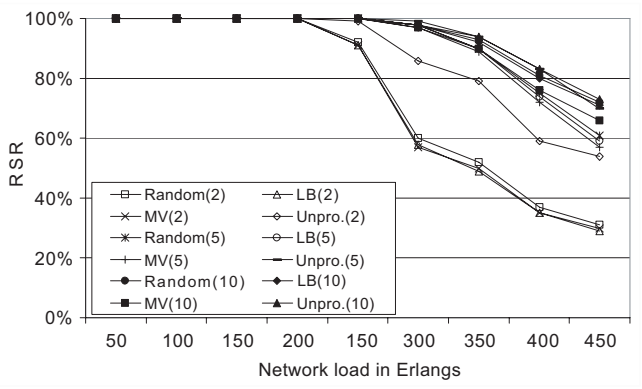


Fig. 8. Reprovisioning success rate (RSR) for MBR for unprotected and vulnerable connections under different policies for MAS=2, 5, 10.

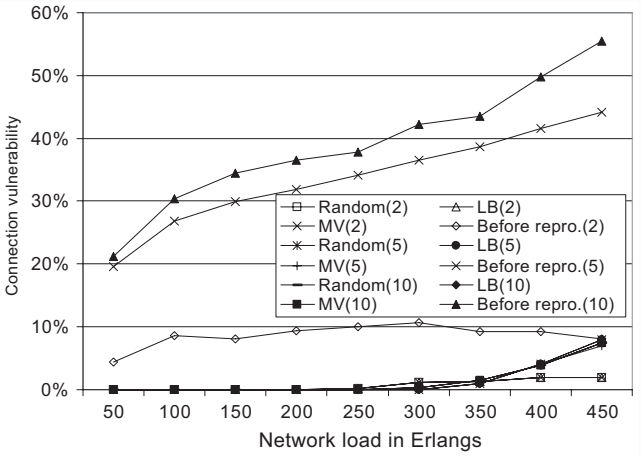


Fig. 9. Connection vulnerability for MBR before and after reprovisioning under different policies for MAS=2, 5, 10.

large. This is because a backup path tends to prefer a longer route to share with existing backup wavelengths (shown by the backup hop distance in Fig. 5) when MAS is large. In both cases (heavy load and large MAS), more connections are sharing the same backup wavelength, so removing one backup wavelength leads to more vulnerable connections.

2) Reprovisioning Approach: Next, we reprovision new backups for unprotected and vulnerable connections using Algorithm 1. Figures 7, 8, and 9 show the results when $MAS =$ 2, 5, and 10. Figure 7 shows the percentage of selected vulnerable connections for reprovisioning over all vulnerable connections under different policies, i.e., Random, LB, and MV. For $MAS = 5$, we find that 10%-25% fewer connections are picked for reprovisioning in MV than in Random and LB. We also observe that the curves decrease with increasing load when the load is less than 300 (the corresponding average link load is 51%), but increase with increasing load when the load is larger than 300. This is because the shareability of the network is small when load is light so successfully reprovisioning one vulnerable connection can only resolve the contention of a small number of connections. When the load increases, network shareability first increases, and then it saturates to MAS, i.e., more (but at most MAS) backup paths are packed together. So, successfully reprovisioning one vulnerable connection can resolve the backup contention of multiple (but at most MAS) connections. Therefore, we need to reprovision fewer vulnerable connections when the load increases (but before the network shareability reaches MAS). After the network shareability saturates at MAS, we need to reprovision a large number of vulnerable connections with increasing load as RSR drops due to limited bandwidth on the links. For $MAS = 2$ and 10, we observe similar trends for the curves, i.e., fewer connections are picked for reprovisioning in MV than in Random and LB, and the curves decrease with increasing load first then increase with increasing load when the network shareability saturates to MAS.

Figure 8 shows the RSR for MBR for unprotected and vulnerable connections using different policies. For $MAS =$ 5, we find that we can achieve 100% successful reprovisioning when load is less than 250 (the corresponding average link load is 42%) for both unprotected and vulnerable connections in our example network, but the success rate drops when the link load is larger than 42% due to limited bandwidth. We compute connection vulnerability after reprovisioning and find that it is significantly reduced, when compared to the connection vulnerability before reprovisioning, as shown in Fig. 9. Examining Figs. 7 and 9, we can see that similar performance can be achieved by all three schemes, but we need to reprovision fewer connections if the MV policy is applied. This demonstrates that selecting connections for

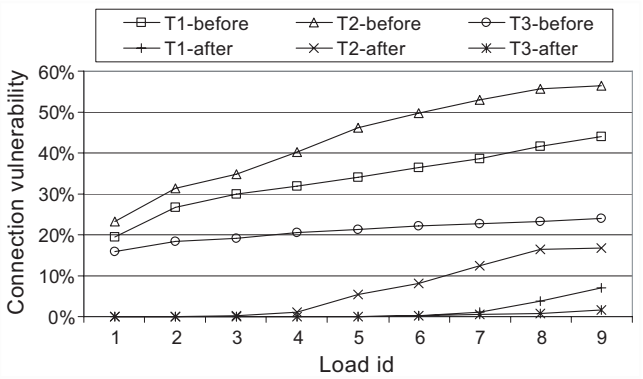


Fig. 10. Connection vulnerability for MBR before and after reprovisioning for three network topologies T_1, T_2, and T_3.

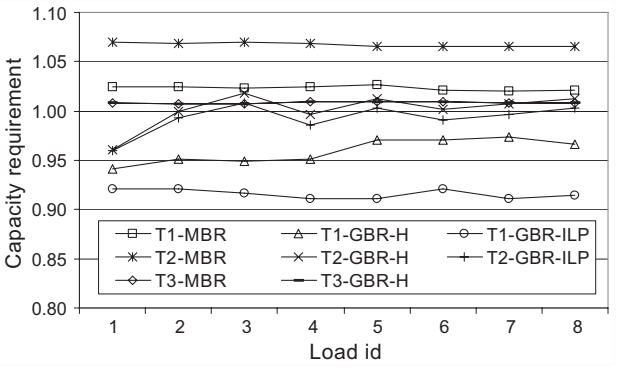


Fig. 11. Capacity requirement for MBR and GBR.

reprovisioning according to the number of violations of a connection is more efficient than according to other selection policies, i.e., Random and LB. In Fig. 8, we also observe that RSR is lower for smaller MAS value. This is because MAS puts a limit on the number of connections that can share a backup wavelength. Thus, reprovisioning has relatively larger chance to fail with smaller MAS due to backup resource sharing.

We also tested our reprovisioning algorithms on other network topologies. Figure 10 shows the connection vulnerability before and after reprovisioning under different network loads for three topologies with $MAS = 5$ and the MV policy (as $MAS = 5$ demonstrates a good tradeoff between bandwidth efficiency and connection vulnerability). T_1 is the network shown in Fig. 3; T_2 is also a US nationwide network with 26 nodes and 40 bidirectional links; and T_3 is a randomly-generated network with 50 nodes and 100 bidirectional links. Table I maps the load ids used in Fig. 10 to the actual loads in Erlangs and the corresponding average link loads for each topology. We find that the reprovisioning algorithm can significantly reduce the connection vulnerability in all three networks, even when the network has 70% link load. Note that there are still vulnerable connections after the reprovisioning as bandwidth is constrained on links. Dynamic restoration can be applied to restore traffic for these vulnerable connections when the next failure occurs. By greatly reducing the connection vulnerability through backup reprovisioning, we drastically reduce the workload for dynamic restoration.

B. Results from Global Backup Reprovisioning (GBR)

We compare the capacity requirement and the computational complexity of the MBR approach to that of the GBR approach. For the results presented in this subsection, we fix the number of wavelength channels in the sample networks to 8 to bound the computation time in the ILP but still vary the average link load from 10% to 70%. In order to achieve 100% RSR, extra bandwidth can be utilized in MBR as well as in GBR.

We compute the total capacity (in terms of wavelength links) used for both primaries and backups of all the connections in the network, both before the failure (denoted as C_b) and after the failure, i.e., after the reprovisioning, (denoted as C_a). Then, C_a/C_b is the capacity ratio required for 100% successful reprovisioning. Figure 11 shows the capacity requirement for MBR and GBR under different network loads (for average link load varying from 10% to 70%) for the three network topologies. Table II gives the actual loads used in Fig. 11 and the computational complexity of MBR for the three network topologies. In MBR, $MAS = 5$ and the selection policy is MV, while in GBR, we apply both ILP and heuristic with $S = 200$. Note that, in the GBR heuristic, we also tried different values of S. Results show that the performance can be improved when S increases, but there is no significant improvement when S is beyond 200. Also, the result of ILP for T_3 is not available as the large number of nodes in T_3 increases its computational complexity dramatically.

We find that around $1.0075-1.0700$ (see Fig. 11) capacity is used for 100% RSR in MBR in the three network topologies, which means that only 0.75-7.00% additional capacity is needed, compared to the total capacity used before the failure. Similarly, in GBR, up to 1.78% additional capacity is needed using the heuristic. The numbers vary slightly with topology or load. In addition, the optimal capacity utilization is achieved when backups are rearranged using the ILP approach in GBR. However, even though GBR outperforms MBR in capacity requirement, it sacrifices the computational complexity. In Section IV-B, we mentioned that the MBR algorithm needs to reprovision (at most) $|T|$ connections at a time but, actually, the number of connections that needs to be reprovisioned is far less than this number. In Table II, we show the percentage of selected connections for the MBR approach with 100% RSR. We find that only around 3.97%-15.82% connections are reprovisioned under various loads for the three sample topologies using the MV selection policy, and which can fully resolve the backup contentions. However, in GBR, all the backups of connections, including unprotected, vulnerable, and unaffected connections, are recomputed. To obtain a minimal solution, we also try 200 different connection sequences (i.e., $S = 200$) in the heuristic. In our simulation, it takes several minutes to several tens of minutes (depending on the topologies and loads) for MBR to complete using a computer with a 1.4-GHz Pentium processor and 512-Mbytes RAM, while GBR takes several hours to several tens of hours to complete (depending on the topologies and loads), for both heuristic and ILP approaches. Considering that the MTTR of a failure is usually on the order of a few hours, the MBR

TABLE I
THE LOADS USED IN FIG. 10 FOR TOPOLOGIES T_1, T_2, AND T_3.

	load id	1	2	3	4	5	6	7	8	9
T_1	load in Erlangs	50	100	150	200	250	300	350	400	450
	avg. link load	0.09	0.18	0.26	0.34	0.42	0.51	0.57	0.65	0.71
T_2	load in Erlangs	50	100	150	200	250	300	350	400	450
	avg. link load	0.13	0.24	0.35	0.46	0.55	0.60	0.65	0.69	0.71
T_3	load in Erlangs	235	345	455	565	675	785	895	1005	1115
	avg. link load	0.18	0.24	0.32	0.39	0.46	0.54	0.61	0.68	0.75

TABLE II
THE LOADS USED IN FIG. 11 AND COMPUTATIONAL COMPLEXITY (COM.) OF MBR.

	load id	1	2	3	4	5	6	7	8
T_1	load in Erlangs	15	30	45	60	75	90	105	120
	avg. link load	0.13	0.24	0.34	0.45	0.55	0.61	0.69	0.71
	com. of MBR	7.55%	10.92%	11.99%	13.69%	14.95%	14.61%	15.82%	15.13%
T_2	load in Erlangs	15	30	45	60	75	90	105	120
	avg. link load	0.17	0.33	0.46	0.54	0.61	0.63	0.67	0.69
	com. of MBR	8.69%	11.18%	13.80%	15.25%	15.22%	15.15%	14.96%	15.18%
T_3	load in Erlangs	20	60	100	140	180	220	260	300
	avg. link load	0.08	0.21	0.32	0.43	0.53	0.62	0.69	0.77
	com. of MBR	3.97%	7.94%	8.28%	8.84%	8.79%	8.67%	8.53%	9.89%

TABLE III
CAPACITY REQUIREMENT (CAP.) AND COMPUTATIONAL COMPLEXITY (COM.) FOR MBR AND GBR-H FOR T_1 WITH 32 WAVELENGTH CHANNELS.

load in Erlangs		60	120	180	240	300	360	420	480
avg. link load		0.11	0.21	0.31	0.41	0.50	0.59	0.67	0.73
cap.	MBR	1.0248	1.0301	1.0310	1.0320	1.0340	1.0350	1.0329	1.0300
	GBR-H	0.9556	0.9876	0.9981	1.0020	1.0067	1.0100	1.0089	0.9982
com.	MBR	12.36%	11.32%	10.14%	9.48%	9.17%	9.09%	8.58%	8.22%

approach demonstrates a good tradeoff between complexity and capacity efficiency, compared to the GBR approach.

We also study the performance under different number of wavelength channels in the network. Tables III summarizes the capacity requirement and the computational complexity for T_1 with 32 wavelength channels for MBR and GBR-H. Similarly, in MBR, $MAS = 5$ and the selection policy is MV, while in GBR-H, $S = 200$. (Note that the result for GBR-ILP is not available due to the computational complexity.) One can observe that results in Table III are similar to Fig. 11 and Table II for the capacity requirement and the computational complexity for MBR and GBR-H.

VII. CONCLUSION

We investigated backup-reprovisioning approaches to handle multiple link failures in WDM mesh networks. New backups are reprovisioned after a network failure occurs to protect against the next potential failure. Both the minimal and global backup reprovisioning approaches (denoted as MBR and GBR, respectively) are studied in wavelength-convertible WDM mesh networks. A reprovisioning algorithm for MBR was proposed based on a generalized network model to provide new backups for vulnerable connections without the knowledge of the location of the next failure. In GBR, both ILP and heuristic-based approaches are proposed. The numerical results show that, using MBR, the connection vulnerability can be significantly reduced even when the network is heavily loaded.

In order to successfully reprovision all the unprotected and vulnerable connections, we need to put more bandwidth in the network. We compared the capacity requirement and computational complexity of MBR to that of GBR through numerical examples. MBR demonstrates a good tradeoff between complexity and capacity efficiency to handle multiple concurrent failures.

REFERENCES

[1] B. Mukherjee, "WDM optical networks: progress and challenges," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1810–1824, Oct. 2000.
[2] O. Gerstel and R. Ramaswami, "Optical layer survivability: a services perspective," *IEEE Commun. Mag.*, vol. 38, pp. 104–113, March 2000.
[3] G. Mohan, C. S. R. Murthy, and A. K. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 9, pp. 553–566, Oct. 2001.

[4] M. Clouqueur and W. D. Grover, "Availability analysis of span-restorable mesh networks," *IEEE J. Select. Areas Commun.*, vol. 20, no. 4, pp. 810–821, May 2002.
[5] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *IEEE/OSA J. Lightwave Technology*, vol. 21, no. 4, pp. 870–883, Apr. 2003.
[6] R. Ramamurthy, A. Akyamac, J.-F. Labourdette, and S. Chaudhuri, "Pre-emptive reprovisioning in mesh optical networks," in *Proc. OFC'2003*, pp. 785–787.
[7] H. Choi, S. Subramaniam, and H.-A. Choi, "On double-link failure recovery in WDM optical networks," in *Proc. IEEE INFOCOM'2002*, vol. 2, pp. 808–816.
[8] S. Kim and S. Lumetta, "Evaluation of protection reconfiguration for multiple failures in WDM mesh networks," in *Proc. OFC'2003*, pp. 210–211.
[9] D. Schupke and R. Prinz, "Performance of path protection and rerouting for WDM networks subject to dual failures," in *Proc. OFC'2003*, pp. 209–210.
[10] S. S. Lumetta and M. Medard, "Towards a deeper understanding of link restoration algorithms for mesh networks," in *Proc. IEEE INFOCOM'2001*, vol. 1, pp. 367–375.
[11] S. S. Lumetta, M. Medard, and Y.-C. Tseng, "Capacity versus robustness: a tradeoff for link restoration in mesh networks," *IEEE/OSA J. Lightwave Technol.*, vol. 18, no. 12, pp. 1765–1775, Dec. 2000.
[12] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," in *Proc. IEEE INFOCOM'2001*, vol. 2, pp. 699–708.
[13] X. Su and C.-F. Su, "An online distributed protection algorithm in WDM networks," in *Proc. ICC'2001*, vol. 5, pp. 1571–1575.
[14] C. Ou, J. Zhang, H. Zang, L. H. Sahasrabuddhe, and B. Mukherjee, "New and improved approaches for shared-path protection in WDM mesh networks," *IEEE/OSA J. Lightwave Technol.*, vol. 22, pp. 1223–1232, May 2004.
[15] C. Lo and B. Chuang, "A novel approach of backup path reservation for survivable high-speed networks," *IEEE Commun. Mag.*, vol. 41, pp. 146–152, March 2003.
[16] S. Gowda and K. M. Sivalingam, "Protection mechanisms for optical WDM networks based on wavelength converter multiplexing and backup path relocation techniques," *Proc., IEEE INFOCOM'2003*, vol. 1, pp. 12–21.
[17] V. Anand and C. Qiao, "Static versus dynamic establishment of protection paths in WDM networks, Part I," *J. High Speed Networks*, vol. 10, no. 4, pp. 317–327, Nov. 2001.

Jing Zhang received the B.S. degree from Peking University, Beijing (China) in 1998 and the M.S. and Ph.D. degree from University of California, Davis (USA), in December 2001 and January 2005, respectively. Her research interests include fault management, algorithm design, performance evaluation, and reliability analysis in communication networks. Currently, she is a performance engineer at Sun Microsystems, Inc.

Keyao Zhu received the B.S. degree from Peking University, Beijing (China) in 1998 and the M.S. and Ph.D. degree from University of California, Davis (USA), in July 2000 and September 2003, respectively.

From August 2003 to September 2004, he was with Research and Innovation, Alcatel Shanghai Bell. Currently, he is a software engineer at Brion Tech. Inc., Santa Clara, CA, USA. Dr. Zhu has served as a Technical Committee Member of ICC'04 and ICC'05. In June, 2004, He received the Zuhair A. Munir Award for the Best Doctoral Dissertation in College of Engineering, University of California, Davis, for his research on WDM Optical Networks.

Biswanath Mukherjee received the B.Tech. (Hons) degree from Indian Institute of Technology, Kharagpur (India) in 1980 and the Ph.D. degree from University of Washington, Seattle, in June 1987. At Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship. In July 1987, he joined the University of California, Davis, where he has been Professor of Computer Science since July 1995, and Chairman of Computer Science since September 1997. He is co-winner of paper awards presented at the 1991 and the 1994 National Computer Security Conferences. He serves on the editorial boards of the *IEEE/ACM Transactions on Networking*, *IEEE Network*, *ACM/Baltzer Wireless Information Networks (WINET)*, *Journal of High-Speed Networks*, *Photonic Network Communications*, and *Optical Network Magazine*. He also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society. He served as the Technical Program Chair of the IEEE INFOCOM '96 conference. Biswanath Mukherjee is author of the textbook *Optical Communication Networks*, published by McGraw-Hill in 1997, a book which received the Association of American Publishers, Inc.'s 1997 Honorable Mention in Computer Science. His research interests include lightwave networks, network security, and wireless networks.