A Practical Approach to Operating Survivable WDM Networks

Murari Sridharan, *Student Member, IEEE*, Murti V. Salapaka, and Arun K. Somani, *Fellow, IEEE*

***Abstract*—Several methods have been developed for joint working and spare capacity planning in survivable wavelength-division-multiplexing (WDM) networks. These methods have considered a static traffic demand and optimized the network cost assuming various cost models and survivability paradigms. Our interest primarily lies in network operation under dynamic traffic. We formulate various operational phases in survivable WDM networks as a single integer linear programming (ILP) optimization problem. This common framework avoids service disruption to the existing connections. However, the complexity of the optimization problem makes the formulation applicable only for network provisioning and offline reconfiguration. The direct use of this method for online reconfiguration remains limited to small networks with few tens of wavelengths.**

Our goal in this paper is to develop an algorithm for fast online reconfiguration. We propose a heuristic algorithm based on LP relaxation technique to solve this problem. Since the ILP variables are relaxed, we provide a way to derive a feasible solution from the relaxed problem. The algorithm consists of two steps. In the first step, the network topology is processed based on the demand set to be provisioned. This preprocessing step is done to ensure that the LP yields a feasible solution. The preprocessing step in our algorithm is based on: a) the assumption that in a network, two routes between any given node pair are sufficient to provide effective fault tolerance and b) an observation on the working of the ILP for such networks. In the second step, using the processed topology as input, we formulate and solve the LP problem. Interestingly, the LP relaxation heuristic yielded a feasible solution to the ILP in all our experiments. We provide insights into why the LP formulation yields a feasible solution to the ILP. We demonstrate the use of our algorithm on practical size backbone networks with hundreds of wavelengths per link. The results indicate that the run time of our heuristic algorithm is fast enough (in order of seconds) to be used for online reconfiguration.

***Index Terms*—Heuristic, integer linear programming (ILP), linear programming (LP) relaxation, optimization, protection, restoration, survivability, wavelength-division multiplexing (WDM).**

I. INTRODUCTION

A. Background

THE EXPLOSIVE growth of Web-related services over the Internet is creating a growing demand for bandwidth. Recent times have witnessed significant shifts in traffic patterns. Major carriers in the United States announced that data traffic, for the first time, has overtaken voice traffic. With deregulation of telecommunication markets in Europe and Asia, the pattern is similar, if not more significant. Many of today's businesses rely heavily on a reliable and continuously available high-speed communications infrastructure. With millions of wavelength miles laid out in typical global and nationwide networks, fiber-optic cables are among the most prone to failures. TEN (formerly Hermes Europe Railtel), a pan-European carriers' carrier network, estimates an average of one cable cut every four days on their network [1]. Therefore, it is imperative to design survivable networks to avoid catastrophic socioeconomic effects.

Today's Internet is dominated by applications and services based on the ubiquitous Internet protocol (IP). The trend is likely to continue as IP continues to provide a form of protection and restoration by enabling packets to be dynamically rerouted around link or node failures. With transmission control protocol providing a reliable transport service, it is very likely that IP-based applications will continue to dominate Internet traffic for years to come. It is therefore evident that the WDM backbone networks should be optimized for IP services. These factors make it attractive to carry fast growing IP traffic directly over an optical network without the intervening synchronous optical network (SONET)/synchronous digital hierarchy (SDH) layer.

SONET has its own protection schemes providing fast recovery (50 ms). The SONET restoration time is dictated by the fact that voice calls could be dropped if the restoration times were any longer. The study in [2] provides interesting comparisons on the impact of restoration times on various applications. To accommodate the changing trends, the entire network needs restoration strategies that are different from the conventional SONET-like implementations [3]. Optical layer protection and restoration offers several advantages.

1) Recovery mechanisms provided by the optical layer are expected to be faster compared to those provided by the higher level service layers.
2) Optical layer can better optimize resources such as wavelengths.
3) Optical layer provides protection to higher layer protocols that do not have their own recovery mechanisms.

Restoration could be provided by either the service layer or the optical layer, and the relative benefits are being debated. However, carriers have made huge investments, and the transition to a single optical layer providing all the services is expected to be slow. Considerable research is needed to understand the interactions of recovery protocols that operate at multiple layers

Manuscript received February 22, 2001; revised July 31, 2001. This work was supported in part by the National Science Foundation under Grant ANI 9973102, in part by the Defense Advanced Research Projects Agency (cofunded by the National Security Agency) under Grant N66001-00-1-8949, and in part by the National Science Foundation under Grant ECS 9733802.

The authors are with the Dependable Computing and Networking Laboratory, Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: murari@iastate.edu; murti@iastate.edu; arun@iastate.edu).

Publisher Item Identifier S 0733-8716(02)00153-1.

in the event of a fiber cut. The outage duration in the event of a failure could be lengthened, as recovery protocols from various layers may interfere with each other. The network could enter into a deadlocked state, never converging to a new topology. As a result, protection interoperability studies for optical networks are gaining considerable importance in the research community [4], [5].

B. Objective

Several methods have been proposed for joint working and spare capacity planning in survivable WDM networks. These methods have considered a static traffic demand and optimized the network cost assuming various cost models and survivability paradigms. Our interest is primarily in network operation under dynamic traffic. In [6] and [7], we captured various operational phases in survivable WDM networks as a single integer linear programming (ILP) optimization problem. This common framework incorporates service disruption. However, the complexity of the optimization problem makes the formulation applicable only for network provisioning and offline reconfiguration. The direct use of this method for online reconfiguration remains limited to small networks with few tens of wavelengths.

C. Outline of this Paper

The remainder of Section I reviews prior work on survivable WDM network design. Section II explores choices for restoration architectures and introduces the restoration model adopted for our formulation. The ILP formulation for various operational phases in survivable WDM networks appears in [6] and [7] and is presented in Section III for convenience. Section IV discusses techniques to prune the size of the ILP. In Section V, we present a heuristic algorithm based on the LP relaxation technique. Section VI compares the ILP and the LP solution for small problem instances and demonstrates the heuristic algorithm applied to large problem instances. Section VII presents our conclusions and observations on the heuristic algorithm.

D. Related Work

To date, design problems in mesh-survivable WDM network research have been studied in [8]–[15]. The study in [8] proposes an optimal design scheme for survivable WDM transport networks in which fast restoration can be achieved by using predetermined restoration paths. Integer programming-based design problems were formulated to optimally determine working paths together with their corresponding restoration paths, the number of fibers in each span, and the optical cross-connects in each node. The study in [9] examines different approaches to protect mesh-based WDM optical networks from single-link failures. ILPs were formulated to determine the capacity requirements for a static traffic demand based on path/link protection/restoration survivability paradigms. In [10], ILP and simulated annealing (SA) were used to solve optimization problems for routing, planning of working capacity, rerouting, and planning of spare capacity in WDM networks. The purpose of the study was to design a fiber topology and optical path layer for WDM networks, with a fixed channel plan, minimizing the total cost for a given traffic demand. The work in [11] aims at providing design protection that is well adapted to WDM networks, where many channels share the same fiber. The design protection, however, does not guarantee carrying all the traffic that was carried prior to the failure. Instead, it aims at maintaining connectivity between all pairs of network ports following a single failure and lets the higher level network layers reconfigure themselves to carry only the high-priority traffic. Joint optimization of primary and restoration routes to minimize the network capacity was studied in [12]. The study also tried to determine the best restoration route for each wavelength demand, given the network topology, the capacities, and primary routes of all demands. The work in [13] mainly concerns connection provisioning for optical networks. A heuristic algorithm was developed for routing and wavelength assignment for a set of static connections, and an adaptation of the algorithm was proposed to handle a set of failures. The study in [14] proposes a methodology for performing automatic protection switching in optical networks with arbitrary topologies in order to protect the network from fiber link failures. The work in [15] studies the influence of modularity and economy-of-scale effects on the survivable network design. Results indicate that there are worthwhile savings by including modularity aspects directly in the design formulation. The significant research finding is the topology reduction arising spontaneously in optimized designs under the combined effects of high modularity and economy-of-scale.

Once the network is provisioned, the critical issue is how to operate the network in such a way that the network performance is optimized under dynamic traffic. Every connection request between a given source–destination (s–d) pair is satisfied by establishing a primary lightpath and a backup (restoration) lightpath. The operational phases in survivable WDM networks are formulated as a single ILP problem [6], [7]. The initial call setup phase is a static optimization problem where the network capacity is optimized, given the topology and a traffic matrix to be provisioned on the network. Once calls arrive dynamically in the network, they are admitted based on a routing and wavelength assignment algorithm. The network cannot afford to run optimization procedures to route every call arriving dynamically. It may be feasible to apply such procedures on a per-call basis for long-term connections in the absence of automated provisioning. However, this trend is expected to change, and provisioning of lightpaths is expected to happen in real time. As a result, for such short-term on-the-fly requests, it may not be feasible to run optimization procedures for every call arriving in the network. As a result, the network capacity utilization slowly degrades to a point when calls may be blocked due to inefficient usage of network resources. This triggers various reconfigurations stages, which try to better utilize the network capacity.

In short- and medium-term reconfiguration, the goal is to optimize resource consumption for a few or all of the backup paths, respectively, while not disturbing the primary paths of the currently working connections. Backup paths are activated only when the primary path fails, so reconfiguring backups does not affect the service. If further optimization is required, a long-term reconfiguration is triggered. The goal of the long-term reconfiguration stage is as follows: given the current working demands in the network and a new set of demands to be provisioned, optimize the network capacity while avoiding service disruption to the current working connections. We use the terms “demands” and “connections” interchangeably in the paper.

The above optimization problem can also be treated as a static formulation that optimizes the network capacity for the complete demand set, which includes the current working demands and the new demands. This treatment provides the best capacity optimization, but all the current connections may be disrupted, which may not be acceptable. We should have an additional requirement to avoid disrupting any of the currently working demands. One way to achieve this is to remove the capacity used by the primary paths of the current working demands and optimize the network capacity only for the new demands. This treatment avoids disruption to the current working paths and tries to optimize on the reduced capacity. This, however, may result in poor capacity utilization. To the best of our knowledge, none of the existing methods captures service disruption in the problem formulation. We avoid service disruption in our framework by adding a penalty term for disrupting existing connections, as explained in Section III.

E. Outline of the Proposed Work

The complexity of the optimization problem makes the formulation applicable only for provisioning and offline reconfiguration. The use of these methods for online reconfiguration is limited to small networks with a few tens of wavelengths. To reduce the complexity of such problems, and to make it more tractable, various decomposition techniques based on Lagrangean relaxation [16] and LP relaxation [17] exist in the literature. We discuss some approaches that have been used for solving large instances of problems in optical networks.

In [12], the Lagrangean relaxation method was used to simplify the integer problem into subproblems for each demand. Since a solution to the relaxed problem may not necessarily be a feasible solution to the original problem, heuristics were employed to extract a feasible solution. The LP relaxation of the ILP model is one of the widely used relaxation techniques. In this technique, the integrality constraints of the ILP variables are relaxed. In [18], the LP relaxation technique was used to derive an upper bound on the carried traffic of connections for any routing and wavelength assignment algorithm. In [19], a randomized rounding technique was used to convert fractional flows provided by the LP solution to integer flows, and graph coloring algorithms were used to assign wavelengths to the lightpaths. The problem of minimizing the total wavelength mileage, in a network with arbitrary topology, to provide shared line protection was studied in [20]. In [21], the authors proposed an efficient approach for solving the wavelength mileage problem. The algorithm provides a feasible solution, with minimal violation of the design constraints, and a pruning technique of the search space to reduce the problem complexity.

Our goal in this paper is to develop an algorithm for fast online reconfiguration. We propose a heuristic algorithm based on the LP relaxation technique to solve this problem. Since the ILP variables are relaxed, we provide a way to derive a feasible solution from the relaxed problem. The algorithm consists of two steps. In the first step, the network topology is processed based on the demand set to be provisioned. This preprocessing step is done to ensure that the LP yields a feasible solution. The preprocessing step in our algorithm is based on a) the assumption that in a network, two routes between any given node pair are sufficient to provide effective fault tolerance, and b) an observation on the working of the ILP for such networks. In the second step, using the processed topology as input, we formulate and solve the LP problem. Interestingly, the LP relaxation heuristic yielded a feasible solution to the ILP in all our experiments. We provide insights into why the LP formulation yields a feasible solution to the ILP. We demonstrate the use of our algorithm on practical size backbone networks with hundreds of wavelengths per link. The results indicate that the run time of our heuristic algorithm is fast enough (in order of seconds) to be used for online reconfiguration.

II. Restoration Architecture

A. Choices for Restoration Architecture

Several survivability paradigms have been explored for surviving single link failures in mesh-based networks [3], [8], [9], [12], [22], [23]. They can be classified based on their route computation and execution mechanisms as centralized or distributed, by their rerouting as path or link based, by their computation timing as precomputed or real time, and their capacity sharing as dedicated or shared. Link-based restoration methods reroute disrupted traffic around the failed link, while path-based rerouting replaces the whole path between the source and destination of a demand. The link-based approach requires the ability to identify a failed link at both its ends and makes restoration difficult when node failures happen. The choice of restoration paths is limited and thus may use more capacity than required. The precomputed approach calculates restoration paths before a failure happens, and the real-time approach does so after the failure occurs. The former approach allows fast restoration as the paths are precomputed, while the latter approach is slow, as the alternate path is computed after the failure is detected. Centralized restoration methods compute primary and restoration paths for all demands at a central controller where current information is assumed to be available. The paths are then downloaded into each node’s route tables. These algorithms are usually path based. They may use precomputed routes or detect routes at run time. As explained above, since this step needs to identify failure, ascertain the remaining topology and capacity, and then find the best alternate route for the affected demands, the procedure is very slow. Given the importance of restoration speed and potential difficulty in fast failure isolation in optical networks, this approach is therefore not very attractive. Centralized schemes that involve precomputed routes are more conducive for practical implementations. However, maintaining up-to-date information requires frequent communications between the nodes and the central controller. This overhead becomes a potential problem as the network size grows. Distributed methods may involve precomputed tables of routes and discovers capacity in real time. Real-time capacity discovery is slow and the capacity utilization may be inefficient. Distributed precomputation of restoration route is an attractive approach. Capacity sharing among the primary and restoration paths can be dedicated or shared. The dedicated technique uses 1:1 protection, where each primary path has a corresponding restoration path. In the shared case, several primaries can have the same backup path as long as the primaries are node and link disjoint. This scheme is called the backup multiplexing technique [22]. These paradigms serve as a good framework for analyzing the different design methodologies, as each design methodology uses a restoration model, which is a combination of the different paradigms just described.

B. Restoration Model

We consider 100% restoration guarantee for any single node or link failure for protected connections. This means that primary and restoration paths of protected connections are allocated the same capacity and are node and link disjoint. We employ the backup multiplexing technique to improve the wavelength utilization. This technique allows many restoration paths, belonging to demands of different node pairs, to share a wavelength λ on link l if and only if their corresponding primary paths are link and node disjoint. It should be noted that although every primary lightpath has a corresponding backup lightpath dedicated to it, wavelengths on a link can be shared by restoration paths belonging to demands of different node pairs, as long as their primary paths do not share any common links. This is due to the fact that no single failure will cause two primary paths to contend for the same backup capacity. This improves wavelength utilization while providing 100% guarantee under the single fault assumption. We have the following constraints in our restoration model.

1) The number of connections (lightpath) on each link is bounded.
2) Demand constraints: All demands both primary and backup for every node pair must be met.
3) Primary path wavelength restrictions: Only one primary path can use a wavelength λ on link l; no restoration path can use the same λ on link l.
4) Restoration path wavelength restrictions: Many restoration paths can share a wavelength λ on link l if and only if their corresponding primary paths are link and node disjoint.
5) Primary and backup paths for a given demand should be node and link disjoint.

III. Formulation of the Optimization Problem

For the ILP formulation, the following information is assumed to be given: the network topology, a demand matrix consisting of the new connections to be established, and the set of current working connections. We also assume that two alternate routes between each node pair are precomputed and given. Information regarding whether any two given routes are link and node disjoint are also assumed to be given. Each route between every s–d pair is viewed as W wavelength continuous paths (lightpaths), one for each wavelength, and therefore, we do not have an explicit constraint for wavelength continuity. The ILP solution determines the primary and backup lightpaths for the demand set and hence determines the routing and wavelength assignment.

A. Notation

The network topology is represented as a directed graph $G(N, L)$ with N nodes and L links with W wavelengths on each link. We also assume that two alternate routes, which are node and link disjoint, for each s–d pair, are used to provide survivability. The following notation is used.

• $n = 1, 2 \ldots, N$	Number assigned to each node in the network.
• $l = 1, 2 \ldots, L$	Number assigned to each link in the network.
• $\lambda = 1, 2 \ldots, W$	Number assigned to each wavelength.
• $i, j = 1, 2 \ldots, N * (N-1)$	Number assigned to each s–d pair.
• $K = 2$	Alternate routes between every s–d pair.
• $p, r = 1, 2, \ldots, KW$	Number assigned to a path for each s–d pair. A path has an associated wavelength (lightpath). Each route between every s–d pair has W wavelength continuous paths. The first $1 \leq p, r \leq W$ paths belong to route 1 and $W+1 \leq p, r \leq 2W$ paths belong to route 2.
• $\bar{p}, \bar{r} = 1, 2, \ldots, KW$	if $1 \leq p, r \leq W$ (route 1), then $W+1 \leq \bar{p}, \bar{r} \leq 2W$ (route 2) and vice versa.
• (i, p)	pth path for s–d pair i.
• d_i	Demand for node pair i, in terms of number of lightpath request. Each request is assigned a primary and restoration route.

The following cost parameters are employed:

• C_l cost of using link l (data);

• C_w cost of disrupting a currently working path (data).

The following information is given regarding whether two given paths are link and node disjoint.

• $I_{(i,p),(j,r)}$ takes a value of one if paths (i, p) and (j, r) have at least one link in common; zero otherwise. If two routes share a link, then all lightpaths using those routes have the corresponding I value set to one; else zero. (data).

The following notations are used for path-related information.

• $\delta^{i,p}$	Path indicator that takes a value of one if (i, p) is chosen as a primary path; zero otherwise (binary variable).
• $\nu^{i,r}$	Path indicator that takes a value of one if (i, r) is chosen as a restoration path; zero otherwise (binary variable).
• $\epsilon_l^{i,p}$	Link indicator that takes a value of one if link l is used in path (i, p); zero otherwise (data).
• $\psi_\lambda^{i,p}$	Wavelength indicator that takes a value of one if wavelength λ is used by the path (i, p); zero otherwise (data).

- $g_{l,\lambda}$ Takes a value one if wavelength λ is used by some restoration route that traverses link l (binary variable).
- $\chi^{i,p}$ Path indicator that takes a value of one if (i,p) is a currently working primary path; zero otherwise (data). We are only interested in the primary paths of the current working connection, as the restoration paths can be reassigned.

B. Problem Formulation

1) Objective: The objective is to minimize the network capacity. The first term in the objective function [(1)] denotes the capacity consumed by primary paths, and the second term denotes the capacity consumed by backup paths. The last term is a penalty term. If a currently working connection $(\chi^{i,p}=1)$ is reassigned in the final solution $(\delta^{i,p}=0)$, then the objective value is penalized by adding a cost C_w to it.
Minimize

$$\sum_{i=1}^{N(N-1)}\sum_{p=1}^{KW}\delta^{i,p}\sum_{l=1}^{L}\epsilon_l^{i,p}C_l+\sum_{l=1}^{L}\sum_{\lambda=1}^{W}g_{l,\lambda}C_l + \sum_{i=1}^{N(N-1)}\sum_{p=1}^{KW}\chi^{i,p}(1-\delta^{i,p})C_w. \quad (1)$$

2) Restoration Path Wavelength Usage Indicator Constraint: $g_{l,\lambda}$ takes a value of one if wavelength λ is used by some restoration route (i,r) that traverses link l. Constraints (3) and (4) set $g_{l,\lambda}=1$, if $X_{l,\lambda}\geq 1$. $X_{l,\lambda}$ counts the number of paths using link l and wavelength λ for backup

$$X_{l,\lambda}=\sum_{i=1}^{N(N-1)}\sum_{r=1}^{KW}\nu^{i,r}\epsilon_l^{i,r}\psi_\lambda^{i,r} \quad (2)$$
$$g_{l,\lambda}\leq X_{l,\lambda} \quad (3)$$
$$N(N-1)WKg_{l,\lambda}\geq X_{l,\lambda} \quad (4)$$
$$1\leq l\leq L,\quad 1\leq\lambda\leq W,\quad X_{l,\lambda}\geq 0.$$

3) Link Capacity Constraint:

$$\sum_{i=1}^{N(N-1)}\sum_{p=1}^{KW}\delta^{i,p}\epsilon_l^{i,p}+\sum_{\lambda=1}^{W}g_{l,\lambda}\leq W\quad 1\leq l\leq L. \quad (5)$$

The demand constraints for each node pair are

$$\sum_{p=1}^{KW}\delta^{i,p}=d_i\quad 1\leq i\leq N(N-1) \quad (6)$$
$$\sum_{r=1}^{KW}\nu^{i,r}=d_i\quad 1\leq i\leq N(N-1). \quad (7)$$

4) Primary Path Wavelength Usage Constraint: Only one primary path can use a wavelength λ on link l; no restoration path can use the same λ on link l

$$\sum_{i=1}^{N(N-1)}\sum_{p=1}^{KW}\delta^{i,p}\epsilon_l^{i,p}\psi_\lambda^{i,p}+g_{l,\lambda}\leq 1\quad 1\leq l\leq L, 1\leq\lambda\leq W. \quad (8)$$

5) Backup Multiplexing Constraint: If $I_{(i,p),(j,r)}$ is one, then only one of the restoration paths can use a wavelength λ on a link l as backup among the primaries contending for backup

$$\left(\nu^{i,p}\epsilon_l^{i,p}\psi_\lambda^{i,p}+\nu^{j,r}\epsilon_l^{j,r}\psi_\lambda^{j,r}\right)I_{(i,\bar{p}),(j,\bar{r})}\leq 1$$
$$1\leq i,\quad j\leq N(N-1),\quad 1\leq p,\bar{p},r,\quad \bar{r}\leq KW. \quad (9)$$

6) Constraint for Topological Diversity of Primary and Backup Paths: Primary and restoration paths of a given demand should be node and link disjoint

$$\sum_{p=1}^{W}\delta^{i,p}=\sum_{r=W+1}^{KW}\nu^{i,r} \quad (10)$$
$$\sum_{p=W+1}^{KW}\delta^{i,p}=\sum_{r=1}^{W}\nu^{i,r}. \quad (11)$$

The ILP can be used in different phases of network operation by appropriately setting the C_w value. For example, in the initial call setup phase, all $\chi^{i,p}s$ are zero, as there are no working connections. Hence the third term in (1) is zero. The higher the value of C_w, the more the guarantee that primary paths of the working connections will remain unaffected. In the short/medium reconfiguration phase, the cost of C_w is typically set very high for the primary paths of the working connections. It is to be noted here that a high value of C_w does not guarantee that the primary path will not be rerouted in the final solution. Hence to avoid disruption to primary paths of working connections, the capacity consumed by them should be removed and the backup capacity consumption can be optimized. In the long-term reconfiguration phase, an intermediate value of C_w is chosen to capture the tradeoff between possibly disrupting all connections and avoiding disrupting any connection.

The number of variables $\delta^{i,p}$ and $\nu^{i,p}$ grow rapidly with network size. This effect is more pronounced with an increase in the number of wavelengths. For a network of size $N=14, W=32$, and $K=2$, there are $K*W=2*32$ instances of each variable for every node pair. Since there are $N*(N-1)=182$ node pairs, we have $11\,648\delta^{i,p}$ variables and $11\,648\nu^{i,p}$ variables. The number of equations will be roughly 125 million ($11\,648^2$). These numbers can be obtained by substituting the values of N, K, W in each of the equations in the ILP formulation. Thus the problem is complex even for small networks.

IV. ILP Problem Size Reduction

In this section, we discuss techniques for ILP problem size reduction.

A. Pruning the Variables

As explained in the previous section, the number of variables $\delta^{i,p}$ and $\nu^{i,p}$ grow rapidly with network size. A smarter solution would be to consider only variables that are relevant to the problem at hand. This implies that variables that are zero are removed. If a node pair does not have any demands to be routed between them, then all the variables relating to that node pair are removed.

Again, for a network of size $N = 14, W = 32$, and $K = 2$, there are $K * W = 2 * 32$ instances of each variable for every node pair, and there are $N * (N - 1) = 182$ such node pairs. Since not all demands have demands to be routed between them, we get a reduction of $K * W = 2 * 32$ instances of each pair. We also get a reduction of $K * W = 2 * 32$ equations for each of the constraints, and so if only ten node pairs have demands to be routed between them, we have to deal with only 1320^2 instead of $11\,648^2$ equations.

Further reductions are possible by considering only links that affect the specific instance of demands to be provisioned. For each link not considered, we get a reduction of 248^2 equations in our example. The above discussions suggest that it is necessary to carefully enumerate the constraints.

B. Demand Normalization Technique

Another procedure that results in significant problem size reductions is the demand normalization technique. Since we deal with wavelength continuous request chunks between node pairs and since all demands between every node-pair source and sink at the same nodes, we do not distinguish between each of those requests.

To reduce the solution space, we treat each chunk of requests between every demand pair as one entity. Since the whole network should have a consistent view of each entity, we normalize the demand sets by finding the greatest common divisor for all the demand requests and dividing each demand set by that factor. The capacities on all links are also normalized. This results in a scaled-down version of the original problem, which is less difficult to solve.

Since the capacity on each link is normalized, the number of wavelengths W reduces by a factor of m, where m is the greatest common divisor of the demand sets. Considering the network with $N = 14, W = 32$, and $K = 2$, and if m is, say, 2, the number of variables reduces by a factor of two and we are left with 660^2 equations, which is a $1/m^2$ reduction. This technique can yield considerable reduction if m were to be comparable to W. An appropriate procedure that can be adopted here is to adjust demand requests to obtain an m comparable to W, and the solution can be adjusted accordingly. If the solution is obtained for more demands than required, then resources may be reclaimed. If fewer demands were given, then the ILP can be solved again with the solution from the previous stage fed in as currently working connections. Such approaches may deviate from the optimal value, but a feasible solution can be obtained. We use this multistage approach in our heuristic, as will be discussed in Section V-C. It is to be noted that the demand normalization technique tends to group demands belonging to a node pair and forces the demand chunks to follow the same route in the network. This may be restrictive, as a more optimal solution in terms of capacity utilization may be obtained if these demands could be spread across multiple diverse routes between the source–destination pair.

Although we employ all the reduction techniques discussed above to get significant reductions, it is not sufficient to guarantee that large networks with a huge number of demands can be solved. Therefore, we still need a faster approach that yields a feasible solution.

V. Heuristic Based on LP Relaxation

In this section, we present a heuristic algorithm based on the LP relaxation. The LP relaxation of the ILP model is one of the most widely used decomposition techniques. Since the integrality constraints of the ILP variables are relaxed, we may violate some constraints when the fractional flows of the LP solution are rounded off to integer flows. There is no guarantee of extracting a feasible solution once the LP provides a solution with fractional flows. Hence, we need a way of forcing the fractional flows to integer flows so that the LP formulation yields a feasible solution.

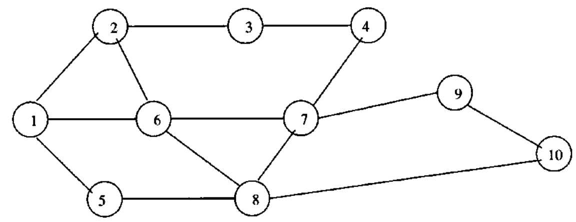
We address this problem using a two-step algorithm. In the first step, the network topology is processed based on the demand set to be provisioned. The preprocessing step is done to ensure that the LP formulation yields a feasible solution. The preprocessing step identifies possible routes for backup multiplexing demands belonging to different node pairs. Each node pair is assigned a set of wavelengths based on a set of rules, as explained in Section V-A.

In the second step, using the processed topology as input, we formulate and solve the LP problem (developed in Section V-B). It is to be noted that the preprocessing step merely assigns a set of wavelengths to each node pair and the actual routing and wavelength assignment is performed by the LP formulation.

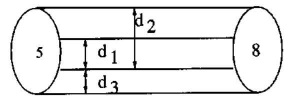
A. Preprocessing Step

The preprocessing is done to ensure that the LP yields a feasible solution, as will be demonstrated later. The preprocessing step in our algorithm is based on a) the assumption that in a network, two routes between any given node pair are sufficient to provide effective fault tolerance, and b) an observation on the working of the ILP for such networks. The ILP in our formulation decides for each node pair which route, of the two available, is to be used for primary and which one for backup. We then classify demands into one of the two categories: a) if two node pairs have common links on both their routes, their backups cannot be multiplexed on the same wavelengths (as they violate the criteria for backup multiplexing), and b) if we have two node pairs and at least one route for each of them is node and link disjoint with the other, then backup paths of demands belonging to these node pairs may or may not be multiplexed depending on the specific instance of traffic that is contending for resources. The preprocessing step is formally presented below.

1) Identify the bottleneck link for each node pair as follows.
   a) Bottleneck link for a node pair i ($Bl[i]$) is defined as that link on either of its two routes, which is part of the routes of most other node pairs.
   b) If multiple links have the same value, the tie is arbitrarily broken.
2) Prewavelength set assignment.
   a) Arbitrarily choose a node pair i.
   b) Assign d_i wavelengths on both its routes. (For satisfying d_i demands, a node pair needs d_i wavelengths on each of its routes for its primary and backup paths. A node pair i is of Type1 on a particular route if it has been assigned exactly the same number of wavelengths as the number of demands on its route). In this case, i is of Type1 on both of its routes.
   c) For every node pair j using $Bl[i]$, we have the following.
      i) One route of i and j already share a common link $Bl[i]$. Without loss of generality, let this be route 1 for both node pairs. Now, if route 2 of node pair j is link and node disjoint with route 2 of i, then assign $d_i + d_j$ wavelengths for j on route 1, out of which d_i wavelengths are shared with i. (A node pair is of Type2 if it has been assigned more wavelengths than its demand requirements.) j is of Type2 in its route 1. On j's other route, it is exactly assigned d_j wavelengths (j is of Type1 on its route 2).
      ii) If node pairs j and i share link(s) on their other route (route 2), then j is assigned d_j wavelengths, disjoint to those assigned to i, on both its routes. In this case, j is Type1 on both its routes.
      iii) Repeat the procedure for all j using $Bl[i]$, comparing with every Type1 node pair available on the link. The following rules apply. (Note: These rules are enforced to handle problems arising as a result of the relaxation. This will be explained in detail in Section V-B).
         A) A Type2 node pair can share wavelengths with only one Type1 node pair on a link.
         B) Every Type1 node pair can have exactly one Type2 pair sharing wavelengths with it. If more than one such Type2 pair exists on the link, for every Type1, then the demands belonging to those node pairs are removed. The problem is solved for only one set of interacting demands at a time.
      iv) Once step 2c) is completed, node pairs that have been assigned wavelengths are marked.
      v) Arbitrarily choose one of the node pairs that has been marked and repeat step 2c) on its bottleneck link.
      vi) Repeat step 2cv) for all marked pairs on link $Bl[i]$.
      vii) Repeat step 2) and terminate when all node pairs that have nonzero demands are marked.



(a)



(b)

Pair	Route 1	Route 2
(1,4)	1–2–3–4	1–5–8–7–4
(2,10)	2–6–8–10	2–1–5–8–7–9–10
(1,3)	1–2–3	1–5–8–7–4–3

(c)

Fig. 1. An illustrative example to demonstrate the preprocessing step.

The preprocessing step identifies possible routes for backup multiplexing demands belonging to different node pairs. Each node pair is assigned a set of wavelengths based on a set of rules [2iiiA) and 2iiiB) above]. It is to be noted that the preprocessing step merely assigns a set of wavelengths to each node pair and the actual routing and wavelength assignment is performed by the LP formulation (developed in Section V-B). We provide insights into the working of the preprocessing step through an illustrative example.

Consider an example network shown in Fig. 1(a). The node pairs of interest to us and the alternate routes between them are shown in Fig. 1(c). Let d_1, d_2, d_3 be the demand request for each pair 1, 2, 3, respectively. The links that are of interest to us are ones where more demands belonging to different node pairs interact. In this example, links $5 \rightarrow 8$, $8 \rightarrow 7$ fall into that category. We arbitrarily choose link $5 \rightarrow 8$ for demonstration. Each node pair is assigned a set of wavelengths. This allocation does not affect the actual routing and wavelength assignment to be performed by the LP formulation. For a node pair, as long as enough wavelengths (capacity) are allocated to meet the demands, it does not matter what range of wavelengths was assigned to it.

Examining link $5 \rightarrow 8$, we arbitrarily choose node pair 3 and assign d_3 wavelengths on both of its routes. Node pair 3 is of Type1 on both of its routes. Next we arbitrarily choose node pair 1, and since node pairs 1 and 3 have common links on both of their routes, they cannot be backup multiplexed with each other. This implies that disjoint wavelength sets are needed for these two link-sharing pairs. Node pair 1 is assigned d_1 wavelengths ($d_3 + 1$ to $\min(d_3 + d_1, W)$ on its route 2) on both of its routes. Hence node pair 1 is also of Type1 on both of its routes. Since node pair 2 has at least one route (its route 1) that is disjoint from the possible routes of both of the other node pairs, there is potential to do backup multiplexing, so the wavelengths may be shared with either of the Type1 pairs available. We arbitrarily choose to share with node pair 1, since a contiguous set of wavelengths can be assigned. The order of wavelengths can be easily rearranged such that contiguous sets of wavelengths can be allocated to Type2 and Type1 node pairs that need to share the same set of wavelengths. Node pair 2 is assigned d_3+1 to $\min(d_3+d_1+d2, W)$ on its route 2 and d_2 wavelengths on its route 1. Hence, node pair 2 is of Type2 in its route 2, as it shares wavelengths with a Type1 node pair 1, and is of Type1 on its route 1. We will assume that the LP solves a demand matrix that is feasible for the ILP.

B. LP Formulation

The LP relaxation of the ILP formulation is developed in this section. In the formulation, $l\min[i,r]$ and $l\max[i,r]$ denote the range of wavelengths assigned for node pair i on routes $r=1,2$.

Minimize

$$\begin{aligned}&\sum_{i=1}^{N(N-1)}\sum_{p=l\min[i,1]}^{l\max[i,1]}\delta^{i,p}\sum_{l=1}^{L}\epsilon_l^{i,p}C_l\\&+\sum_{i=1}^{N(N-1)}\sum_{p=l\min[i,2]}^{l\max[i,2]}\delta^{i,p}\sum_{l=1}^{L}\epsilon_l^{i,p}C_l\\&+\sum_{l=1}^{L}\sum_{\lambda=1}^{W}(-1*g_{l,\lambda})C_l\\&+\sum_{i=1}^{N(N-1)}\sum_{p=l\min[i,1]}^{l\max[i,1]}\chi^{i,p}(1-\delta^{i,p})C_w\\&+\sum_{i=1}^{N(N-1)}\sum_{p=l\min[i,2]}^{l\max[i,2]}\chi^{i,p}(1-\delta^{i,p})C_w.\end{aligned}\tag{12}$$

The demand constraints for each node pair are

$$\sum_{p=l\min[i,1]}^{l\max[i,1]}\delta^{i,p}+\sum_{q=l\min[i,2]}^{l\max[i,2]}\delta^{i,q}=d_i\quad 1\le i\le N(N-1)\tag{13}$$

$$\sum_{p=l\min[i,1]}^{l\max[i,1]}\nu^{i,p}+\sum_{q=l\min[i,2]}^{l\max[i,2]}\nu^{i,q}=d_i\quad 1\le i\le N(N-1).\tag{14}$$

1) Restoration Path Wavelength Usage Indicator Constraint:

$$X_{l,\lambda}=\sum_{i=1}^{N(N-1)}\sum_{r=1}^{KW}\nu^{i,r}\epsilon_l^{i,r}\psi_\lambda^{i,r}\tag{15}$$

$$2g_{l,\lambda}=X_{l,\lambda}\tag{16}$$

$$1\le l\le L,\quad 1\le\lambda\le W$$

$$0\le g_{l,\lambda}\le 1,\quad X_{l,\lambda}\ge 0,\quad 0\le\delta^{i,p},\quad \nu^{i,p}\le 1$$

2) Primary Path Wavelength Usage Constraints:

$$\begin{aligned}&\sum_{i=1}^{N(N-1)}\sum_{p=l\min[i,1]}^{l\max[i,1]}\delta^{i,p}\epsilon_l^{i,p}\psi_\lambda^{i,p}\\&+\sum_{i=1}^{N(N-1)}\sum_{q=l\min[i,2]}^{l\max[i,2]}\delta^{i,q}\epsilon_l^{i,q}\psi_\lambda^{i,q}+g_{l,\lambda}\le 1\\&\qquad 1\le l\le L, 1\le\lambda\le W.\end{aligned}\tag{17}$$

3) Constraint to Ensure That Type2 Primary Never Clashes With Type1 Backups: For Type2 demands on a link l, the following constraint applies. Node pair j belongs to Type1. Node pair i belongs to Type2, which shares wavelengths with node pair j. p, r are those paths on the node pair routes that use $Bl[i]$

$$\nu^{j,r}+\delta^{i,p}\le 1.\tag{18}$$

We demonstrate that the LP yields a feasible solution based on the following observation. This observation is assumed as the basis for further argument. For a given node pair, the LP formulation has a different cost associated with the primary and backup variables. Also, the cost incurred depends only on the route on which the variable's path is present. For a given node pair, if all the LP constraints are being met, the LP will prefer to route the primary variable of a demand on the route that incurs a lesser cost, and as long as the constraints are being met will allocate all primaries on the same route. The same reasoning holds for the backup variables. Thus it can be expected that the primary variables $\delta^{i,p}$ for a particular node pair i take nonzero values only on one route. The same is expected of the backup variables. We state the observation more formally as follows.

Observation 1: The LP has a tendency to group the weights of the variables $\delta^{i,p}$ and $\nu^{i,p}$ for any given i. As a result, for any i, r and $l\min[i,r]\le p\le l\max[i,r]$, either all $\delta^{i,p}$ variables have nonzero assignments or all $\nu^{i,p}$ variables have nonzero assignments.

Based on Observation 1, we make claims that provide insights into why LP formulation yields a feasible solution for the ILP. These claims elucidate the operation of heuristic based on LP relaxation. We now state the claims and provide arguments to support the claims.

Claim 1: The LP solution guarantees integer (binary) assignments for all Type1 variables.

Indeed, consider (13) and (14). They are of the form $A+B=d_i$ and $C+D=d_i$. Terms A, C represent variables on one route and B, D represent variables on the other route. Based on Observation 1, either A or C is zero. Without loss of generality, let the term $C=0$. This would force $D=d_i$ and hence $B=0$. We now have $A=d_i$ and $D=d_i$. Recall that for Type1 variables, $l\max[i,r]-l\min[i,r]=d_i$. Since $0\le\delta^{i,p},\nu^{i,p}\le 1$, all the variables in terms A (primary variables) and D (backup variables) are forced to be assigned as one and all the variables in the other terms are zero.

Claim 2: The LP solution guarantees integer (binary) assignments for all Type2 δ variables.

The above claim follows from the following argument. Let node pair i be of Type1 and j be of Type2. All variables, primary and backup of node pair i, are guaranteed to be binary (Claim 1). Equations (13) and (14) are of the form $A+B=d_j$ and $C+D=d_j$. Terms A, C represent variables on one route and B, D represent variables on the other route. Without loss of generality, let the term $A=0$. This would force $B=d_j, D=0$, and $C=d_j$. Recall that for Type2 variables, $l\max[i,r]-l\min[i,r]=d_j$ on one of its routes and $l\max[i,r]-l\min[i,r]=d_i+d_j$ on its other route. Let B represent variables on the route where d_i+d_j has been assigned. d_i out of d_i+d_j belong to Type1 variables and are guaranteed to be one. Equations (17) and (18) ensure that d_i out of the d_i+d_j variables cannot be used and hence forces the remaining δ variables in term B to be one.

A similar argument can be applied by letting $B=0$. In this case, $A=d_i$ and $l\max[i,1]-l\min[i,1]=d_j$ for A, and hence the δ variables are forced to be one.

A similar argument can be applied for ν variables of Type2 when $C=d_j$ and $l\max[i,1]-l\min[i,1]=d_j$ for that route and ν variables in C are forced to be one. Suppose that $D=d_j$ and $l\max[i,2]-l\min[i,2]=d_i+d_j$; then there are two cases. If d_i were primaries, then (18) forces the variables in D to be one. However, if d_i were backups, then we have d_i+d_j variables and d_j capacity to fill. In this one case, the assignments may be fractional. This case is still acceptable because these violations occur only when Type1 and Type2 backups share the link on the route. Since we allow this case for backup multiplexing, we might be able to reclaim resources by adjusting the fractional flows of Type2 to be one and make it coincide with the backups of Type1.

We now proceed with the LP formulation. In the ILP formulation, $g_{l,\lambda}$ takes a value of one or zero. We should find a way to identify that a wavelength λ is being used as a backup or else (17) will be violated and the primary and backup path may end up using the same wavelength on a link.

We have to appropriately modify $g_{l,\lambda}$ for the LP and make it choose a higher value whenever a wavelength on a link is used for backup. Recall rules 2iiiA and 2iiiB in the heuristic algorithm, which state that every Type1 node pair can have exactly one Type2 pair sharing wavelengths with it. If more than one such Type2 pair exists on the link, for every Type1, then the demands belonging to those node pairs are removed. The problem is solved for only one set of interacting demands at a time. The multistep procedure for such a solution and its implications are discussed in Section V-C. Since only one Type2 demand is allowed to share wavelengths with Type1 demands, the value of $X_{l,\lambda}$, which counts the number of backup paths that share a wavelength λ on link l, can be either zero (if the path is not used for backup), one (one backup path), or two (if two paths share this link l and wavelength λ as backup). Equation (3) of the ILP is modified as shown in (16).

Since $X_{l,\lambda}$ can take values of zero, one, or two (enforced by rules 2iiiA, 2iiiB), $g_{l,\lambda}$ in (16) can take values 0, 0.5, or 1, respectively. In the ILP formulation, $g_{l,\lambda}$ is guaranteed to be one or zero. In the LP formulation, this cannot be captured exactly. Since $g_{l,\lambda}=0.5$ implies that only one backup path uses link l and wavelength λ, $g_{l,\lambda}=1$ implies that two backup paths share link l and wavelength λ, we can modify the objective to make it favor cases when $g_{l,\lambda}=1$. This formulation is not exact, since the cost of two backup paths sharing link l and wavelength λ $(g_{l,\lambda}=1)$ is the same as using two different wavelengths for backup $(2*g_{l,\lambda}=1)$. The modified objective is shown in (12). Equations (5) and (9), representing link capacity constraint and backup multiplexing constraint of the ILP, are no longer constraints in the LP formulation, as these constraints are ensured in the preprocessing step.

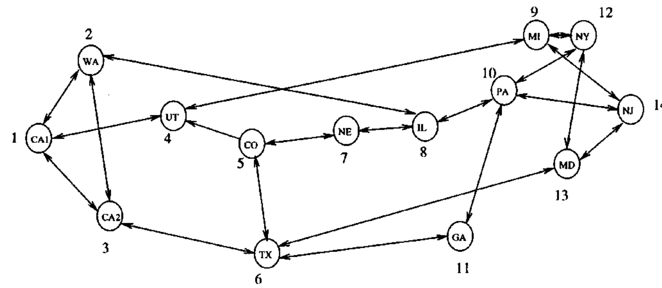


Fig. 2. The 14-node 21-link NSFNET.

C. Solving for Excess Demands

As explained in the previous subsection, every Type1 node pair can have exactly one Type2 pair sharing wavelengths with it. If more than one such Type2 pair exists on the link, for every Type1, then the demands belonging to those node pairs are removed. In such cases, the problem is solved for one set of interacting demands at a time. We propose a multistage approach to solving this problem. We used a similar approach to solve large demands sets in [7] and [24]. At each stage, one instance of the problem is solved, for one set of interacting demands, and the result is used in successive stages. If the problems are solved independently, the resulting solution may be infeasible, as the same path might be used by multiple primaries or backups. To avoid infeasibility, we feed the information about one stage to the next through the $\chi^{i,p}$ variable. Typically, this variable is used to feed information about existing paths to avoid service disruption. We exploit this aspect of our formulation by feeding the solution of one stage to the next. The objective function is modified to include backups chosen during one stage to be fed to the next. This feature is exploited only to make sure that assignments are binary. However, there may be a penalty for this type of solution. First, because the problem is solved sequentially and is not shown the full solution space, the result may be suboptimal. Secondly, depending on the solution from one stage, some demands may be blocked.

VI. Results

A. Experimental Design

We use CPLEX Linear Optimizer 7.0 [25] to solve the ILP and the LP formulations. The experiments were run on a Pentium III 500-MHz processor with 256-MB RAM (note that the solution to the optimization problem is both CPU and memory intensive). These data are provided for the results on run times of our algorithm presented later in this section. We ran our experiments on the 14-node 21-link NSFNET topology (shown in Fig. 2) and the 20-node 32-link ARPANET topology (shown in Fig. 3).

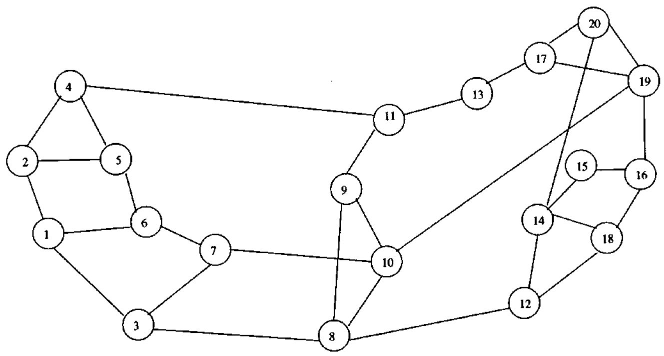


Fig. 3. The 20-node 32-link ARPANET.

TABLE I
ILLUSTRATIVE EXAMPLE

Node pair	Alternate routes
1	1 2
	1 3 2
27	3 1
	3 2 1
110	9 4 5 6
	9 12 13 6
167	13 6 11
	13 12 10 11
32	3 6 5 7
	3 2 8 7

The following experiments are presented in the remainder of this section. We present sample results, that provide insight into the working and quality of our LP formulation, presented in Section VI-B. The complexity of the optimization problem makes the ILP solution intractable for large problem instances. This effect is sometimes seen for small problem instances. In Section VI-C, we compare run times for the ILP and LP solutions for small problem instances. Finally, for large problem instances, we demonstrate the run time of our LP heuristic algorithm in Section VI-D. The LP heuristic algorithm yielded a feasible solution in all the experiments presented in this section.

B. Insights Into the Working and Quality of the LP Heuristic Algorithm

Consider the node pairs and their two alternate routes shown in Table I. Let the number of wavelengths per link be ten. Let the node pairs, in this example, require five primaries and five backups. Since we have a restriction in our model that only one Type2 node pair can share wavelengths with a Type1 node pair on a link, demand requests for node pair 32 are removed in the first stage. The results of the ILP and the LP are shown in Tables II and III, respectively.

The ILP solution assigns backups for demands belonging to node pairs 1 and 27 in the route that has the common link 3→2 and similarly assigns backups for demands belonging to node pairs 110 and 167 on the route that has the common link 13→6. The primary paths were assigned as shown in the Table II. Now, let us compare the LP solution in Table III.

Since the cost of two backup paths sharing a link and wavelength is the same as using two different wavelengths for backup (refer to discussion on $g_{l,\lambda}$ in Section V-B), the backup wavelength assignment is different from the ILP assignment. As in the case of the ILP solution, the backups for demands belonging to node pairs 1 and 27 are assigned on the route that has the common link 3→2. But the wavelength assignment for backups is different. The backup paths for node pair 1 were assigned on route 1→ 3 →2 on wavelengths $\lambda_1 - \lambda_5$, and backups for node pair 27 were assigned on route 3→ 2 →1 on wavelengths $\lambda_5 - \lambda_9$. Only one wavelength (λ_5) was used for backup multiplexing, as against all five $(\lambda_1 - \lambda_5)$ in the ILP solution. However, once the LP provides this feasible solution, we may, in

TABLE II
ILP SOLUTION (FIVE DEMAND REQUESTS/NODE PAIR, 10 W/LINK)

Node pair	Alternate routes	Primary	Backup
1	1 2	$\lambda_1 - \lambda_5$	-
	1 3 2	-	$\lambda_1 - \lambda_5$
27	3 1	$\lambda_1 - \lambda_5$	-
	3 2 1	-	$\lambda_1 - \lambda_5$
110	9 4 5 6	$\lambda_1 - \lambda_5$	-
	9 12 13 6	-	$\lambda_1 - \lambda_5$
167	13 6 11	-	$\lambda_1 - \lambda_5$
	13 12 10 11	$\lambda_1 - \lambda_5$	-
32	3 6 5 7	$\lambda_1 - \lambda_5$	-
	3 2 8 7	-	$\lambda_1 - \lambda_5$

TABLE III
LP SOLUTION (FIVE DEMAND REQUESTS/NODE PAIR, 10 W/LINK)

Node pair	Alternate routes	Primary	Backup
1	1 2	$\lambda_1 - \lambda_5$	-
	1 3 2	-	$\lambda_1 - \lambda_5$
27	3 1	$\lambda_1 - \lambda_5$	-
	3 2 1	-	$\lambda_5 - \lambda_9$
110	9 4 5 6	$\lambda_1 - \lambda_5$	-
	9 12 13 6	-	$\lambda_1 - \lambda_5$
167	13 6 11	$\lambda_1 - \lambda_5$	-
	13 12 10 11	-	$\lambda_1 - \lambda_5$
32	3 6 5 7	$\lambda_1 - \lambda_5$	-
	3 2 8 7	-	$\lambda_1 - \lambda_5$

TABLE IV
LP/ILP SOLUTION (TEN DEMAND REQUESTS/NODE PAIR, 10 W/LINK)

Node pair	Alternate routes	Primary	Backup
1	1 2	$\lambda_1 - \lambda_{10}$	-
	1 3 2	-	$\lambda_1 - \lambda_{10}$
27	3 1	$\lambda_1 - \lambda_{10}$	-
	3 2 1	-	$\lambda_1 - \lambda_{10}$
110	9 4 5 6	$\lambda_1 - \lambda_{10}$	-
	9 12 13 6	-	$\lambda_1 - \lambda_{10}$
167	13 6 11	-	$\lambda_1 - \lambda_{10}$
	13 12 10 11	$\lambda_1 - \lambda_{10}$	-

TABLE V
SAMPLE RESULTS DEMONSTRATING THE QUALITY OF THE LP SOLUTION

Demands	ILP Objective	LP Objective
10	38	43
20	76	90
30	114	120
40	152	156
50	190	190

TABLE VI
COMPARING ILP AND LP SOLUTION RUN TIMES

Demands	ILP Time (in secs) (PT)	ILP Time in secs (FT)	LP Time (in secs)
22	601 (3.35% mipgap)	>9000	0.12
32	3973 (4.40% mipgap)	>9000	0.11
42	852.31 (4.03% mipgap)	>9000	0.13
52	104.87	104.87	0.14
72	84.00	84.00	0.17
92	20.84 (0.29% mipgap)	8289.76	0.23

such cases, merge the backup routes to coincide with backup paths of node pair 1 and reclaim the wavelengths (refer to the discussion in Section V-B on adjusting Type2 backups to coincide with backup paths of its corresponding Type1. In this case, demands of node pair 1 belong to Type1 and those of node pair 27 belong to Type2).

For the next set of node pairs, 110 and 167, primary paths for demands belonging to both pairs were chosen on their first route and backup paths on their second route, as shown in Table III. Hence, no backup multiplexing was done. This is in contrast with the ILP solution that used the route containing the common link 13→6 for routing backups and as a result could backup multiplex the demand requests of node pairs 110 and 167.

In the above example, node pair 32 has to be solved in the next stage. In such cases, the solution from the first stage is fed to the second stage as currently working primary and backup paths. In this example, we considered that since the LP chose the backup routes for node pairs 1 and 27 on the route that uses link 3→2, all the requests for node pair 32 were accommodated with the primary and backup route and wavelength assignments as shown in Table III. Although the demands for node pair 32 were accommodated in this example, there is no guarantee that all the demands will be accepted for node pairs that are solved in successive stages. Thus, there may be a penalty for solving the problem sequentially, as discussed in Section V-C.

Now suppose that the node pairs required ten demands each instead of five demands as in the previous case. The solution for the LP and ILP for this case is the same and is shown in Table IV. The LP in this situation, to accommodate all demand requests, is forced to backup multiplex all possible demands and thus yields an optimal solution. It is well known that if the LP relaxation to the ILP provides a solution that is an integer vector, then the solution is feasible and hence optimal to the ILP [17]. This is the reason for the LP's providing an optimal and a feasible solution to the ILP in this case, as the LP solution vector is forced to be an integer in such cases. This behavior is demonstrated in Table V. The results are run on the NSFNET topology with ten wavelengths per link for the example in Table I, with demand requests distributed uniformly across five node pairs. As explained earlier, the reason why the LP yields an optimal and a feasible solution to the ILP as the number of demand requests per node pair increase (comparable to the capacity on the link) is due to the fact that the LP solution vector is forced to be an integer in such cases.

C. Comparing ILP and LP Solution Run Times

The complexity of the optimization problem makes the ILP solution intractable for large problem instances. This effect is sometimes seen for small problem instances. The ILP and LP solution run-time comparison is shown in Table VI. In the table, PT and FT denote partial and full terminations. CPLEX terminates mixed integer optimizations under a variety of circumstances [25]. CPLEX will find an integer optimal solution and terminates when all nodes have been processed. Optimality in this case is relative to the tolerances and other optimality criteria set by the user. The default relative optimality tolerance is 0.0001, in which case the final integer solution is guaranteed to be within 0.01% (default mipgap in CPLEX) of the optimal value. Many formulations do not require such tight tolerances. Requiring CPLEX to seek integer solutions that meet 0.01% tolerance in such cases is wasted computation time. In our case, to make the comparison fair, we report results when the mipgap is around 3–5%. The problem could be terminated when the mipgap reaches within a desired value. As the results show, the LP solution time is considerably less for this example. In the ILP solution result, the fast run times for 52 and 72 demands as against the slower run times for smaller demand requests is not surprising. The solver performs a lot of preprocessing, and depending on how close the initial solution is to the final integer optimal solution, the problem can run that much faster.

TABLE VII
NUMERICAL RESULTS FOR 14-NODE NSFNET TOPOLOGY WITH 100 WAVELENGTHS PER LINK

Demands	LP Constraints	LP Variables	LP Time (in secs)
100	14029	4280	0.52
150	22029	4520	1.10
200	33229	4760	2.18
250	47629	5000	3.89
300	56429	5160	5.25
400	78829	5480	21.87
500	107629	5800	15.75

TABLE VIII
NUMERICAL RESULTS FOR 20-NODE ARPANET TOPOLOGY WITH 100 WAVELENGTHS PER LINK

Demands	LP Constraints	LP Variables	LP Time (in secs)
100	22117	9880	0.70
200	31767	10360	1.16
300	47817	10840	3.06
400	70267	11320	4.94
500	99117	11800	8.34
600	116767	12120	29.04
700	137617	12440	27.26
800	161667	12760	35.64
900	188917	13080	44.24
1000	219367	13400	30.91

D. Run Times for the LP Heuristic Algorithm

We demonstrate the use of our algorithm on practical size backbone networks with hundreds of wavelengths per link. Numerical results for NSFNET and ARPANET topologies, with 100 wavelengths per link, are shown in Tables VII and VIII, respectively. All the techniques discussed for problem size reduction were applied before the LP was solved. The complexity of the problem is determined by the number of variables and constraints in the formulation. We can see from the results that for large demand sets, the run time of our heuristic algorithm is considerably fast (on the order of seconds). This has a great impact on the applicability of our solution for online decision-making at various phases in survivable WDM network operation.

VII. Conclusion

Considerable literature exists on the design of survivable WDM networks. In this paper, our focus is on network operation under dynamic traffic. Once the network is provisioned, the critical issue is how to operate the network in such a way that the network performance is optimized under dynamic traffic. The various operational phases in survivable WDM networks are formulated as a single ILP optimization problem. This common framework incorporates service disruption. However, the complexity of the optimization problem makes the formulation applicable only for network provisioning and offline reconfiguration. The direct use of this method for online reconfiguration remains limited to small networks with a few tens of wavelengths.

We propose a heuristic algorithm based on an LP relaxation technique to solve this problem. Since the ILP variables are relaxed, we provide a way to derive a feasible solution from the relaxed problem. The algorithm consists of two steps. In the first step, the network topology is processed based on the demand set to be provisioned. This preprocessing step is done to ensure that the LP yields a feasible solution. The preprocessing step in our algorithm is based on: a) the assumption that in a network two routes between any given node pair are sufficient to provide effective fault tolerance and b) an observation on the working of the ILP for such networks. In the second step, using the processed topology as input, we formulate and solve the LP problem.

Interestingly, the LP relaxation heuristic yielded a feasible solution to the ILP in all our experiments. We provide insights into why this is so. The claims we make in this paper are argued based on the following observation. For demands belonging to a given node pair, the optimization formulation has a tendency to group the primary paths of all demands on one route and the backup paths on the alternate route. We provide arguments to support the observation. Currently, we are working on proving the observation, or alternatively providing conditions under which the observation holds true. We are extending our formulation to accommodate pairs of alternate routes for each node pair, and the optimization problem can be made to choose from such candidate pairs. Our relaxation heuristic can directly be used with optimization formulations that provide such sets of candidate pairs for every node pair.

We presented sample results that provide insight into the working and quality of our LP formulation. We showed that as the number of demand requests per node pair increases (comparable to the capacity on the link), the LP yields an optimal and a feasible solution to the ILP, as the LP solution vector is forced to be an integer in such cases. We also provided comparison on the run times of the ILP and the LP solution, and the LP solution run time is considerably fast. We demonstrated the use of our algorithm on practical size backbone networks with hundreds of wavelengths per link. We can see from the results that for large demand sets, the run time of our heuristic algorithm is considerably fast (in order of seconds). This has a great impact on the applicability of our solution for online decision-making at various phases in survivable WDM network operation.

References

[1] P. F. Fonseca, "Pan-european multi-wavelength transport networks: Network design, architecture, survivability and sdh networking," in *Proc. 1st Int. Workshop Reliable Communication Networks*, May 17–20, 1998. Paper P3.

[2] J. Sonosky, "Service applications for sonet dcs distributed restoration," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 59–68, Jan. 1994.

[3] O. Gerstel and R. Ramaswami, "Optical layer survivability—An implementation perspective," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1885–1899, Oct. 2000.

[4] O. Crochat, J. Y. Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 8, pp. 384–395, June 2000.

[5] J. Kroculick and C. Hood, "Provisioning multilayer resilience in multiservice optical networks," in *OptiComm: Optical Networking and Communications*, Oct. 2000, pp. 30–41.

[6] M. Sridharan, M. V. Salapaka, and A. K. Somani, "Operating mesh-survivable WDM transport networks," in *Proc. SPIE Int. Symp. Terabit Optical Networking*, Nov. 2000, pp. 113–123.

[7] M. Sridharan, A. K. Somani, and M. V. Salapaka, "Approaches for capacity and revenue optimization in survivable WDM networks," *J. High Speed Networks (Special Issue on Survivable Optical Networks)*, vol. 10, no. 2, pp. 109–125, Aug. 2001.

[8] Y. Miyao and H. Saito, "Optimal design and evaluation of survivable WDM transport networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1190–1198, Sept. 1998.

[9] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks, part I—Protection," in *IEEE INFOCOM*, vol. 2, Mar. 1999, pp. 744–751.

[10] B. V. Caenegem, W. V. Parys, F. De Turck, and P. M. Demeester, "Dimensioning of survivable WDM networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1146–1157, Sept. 1998.

[11] O. Crochat and J. Y. Boudec, "Design protection for WDM optical networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1158–1165, Sept. 1998.

[12] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration," *Bell Labs Tech. J.*, pp. 58–83, Jan.–Mar. 1999.

[13] M. Alanyali and E. Ayanoglu, "Provisioning algorithms for WDM optical networks," in *IEEE INFOCOM*, vol. 2, Mar. 1998, pp. 910–918.

[14] G. Ellinas, A. G. Hailemariam, and T. E. Stern, "Protection cycles in mesh DWDM networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1924–1937, Oct. 2000.

[15] J. Doucette and W. D. Grover, "Influence of modularity and economy-of-scale effects on design of mesh-restorable DWDM networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1912–1923, Oct. 2000.

[16] M. L. Fischer, "The lagrangean relaxation method for solving integer programming problems," *Manage. Sci.*, vol. 27, no. 1, pp. 1–18, 1981.

[17] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxofrd, U.K.: Oxford Univ. Press, 1995.

[18] R. Ramaswamy and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, pp. 489–500, Oct. 1995.

[19] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 903–908, June 1996.

[20] A. Fumagalli, M. Tacca, I. Cerutti, F. Masetti, R. Jagannathan, and S. Alagar, "Survivable networks based on optimal routing and WDM self-healing rings," in *IEEE INFOCOM*, Mar. 1999, pp. 726–733.

[21] A. Fumagalli, M. Tacca, I. Cerutti, F. Masetti, R. Jagannathan, and A. Lardies, "Effects of design constraints on the total wavelength mileage in optical mesh networks with shared line protection," in *Proc. OptiComm: Optical Networking and Communications*, Oct. 2000, pp. 42–53.

[22] G. Mohan and A. K. Somani, "Routing dependable connections with specified failure restoration guarantees in WDM networks," in *IEEE INFOCOM*, Mar. 2000, pp. 1761–1770.

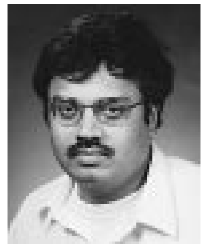
[23] T. H. Wu, *Fiber Network Service Survivability*. Norwood, MA: Artech House, 1992.

[24] M. Sridharan and A. K. Somani, "Revenue maximization in survivable WDM networks," in *Proc. OptiComm: Optical Networking and Communications*, Oct. 2000, pp. 291–302.

[25] *ILOG CPLEX 7.0 Reference Manual* [Online]. Available: http://www.cplex.com

Murari Sridharan (S'99) received the bachelor's degree in computer science and engineering from Sri Venkateswara College of Engineering, University of Madras, Chennai, India, in 1997. He is currently pursuing the Ph.D. degree in computer engineering in the Department of Electrical and Computer Engineering, Iowa State University, Ames.

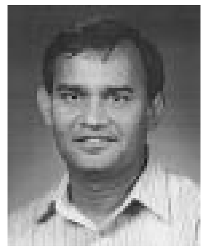
He is a Codeveloper of the Hierarchical Modeling and Analysis Package (HIMAP), a reliability modeling and analysis tool, which is currently being used at Boeing, Honeywell, and several other companies and universities. His research interests include algorithms and architectures for optical networks, fault tolerance, system modeling, and performance analysis.



Murti V. Salapaka was born in Andhra Pradesh, India, on January 30, 1969. He received the B.Tech. degree from the Indian Institute of Technology, Chennai, in 1991 and the M.S. and Ph.D. degrees from the University of California, Santa Barbara, in 1993 and 1997, respectively, all in mechanical engineering.

His research interests include robust control, nanomachining, control of molecular systems and dynamical analysis, control of microcantilever-based devices, and applications of control theory.

Dr. Salapaka received the 1997 NSF CAREER Award.



Arun K. Somani (S'83–M'83–SM'88–F'99) received the M.S.E.E. and Ph.D. degrees in electrical engineering from McGill University, Montreal, PQ, Canada, in 1983 and 1985, respectively.

He is currently David C. Nicholas Professor of Electrical and Computer Engineering at Iowa State University, Ames. He was a Scientific Officer for the government of India, New Delhi, from 1974 to 1982. From 1985 to 1997, he was a Faculty Member at the University of Washington, Seattle, where he was a Professor of electrical engineering and computer science from 1995 onwards. His research interests are in the areas of fault-tolerant computing, computer communication and networks, wireless and optical networking, computer architecture, and parallel computer systems. He has taught courses in these areas and published more than 150 technical papers. He is an Editor of *Microprocessors and Microsystems*.

Professor Somani's has served on program committees of various conferences in his research areas. He was the General Chair of the IEEE Fault Tolerant Computing Symposium and Technical Program Chair of the IEEE Conference on Computer Communications and Networks. He is currently an Associate Editor of IEEE TRANSACTIONS ON COMPUTERS. He is also a Distinguished Lecturer and Distinguished Tutorial Speaker of IEEE.