Dynamic Load Balancing for WDM-based Packet Networks

Aradhana Narula-Tam
MIT Lincoln Laboratory
Lexington, Massachusetts
arad@ll.mit.edu

Eytan Modiano
MIT Laboratory for Information and Decision Systems
Cambridge, Massachusetts
modiano@mit.edu

Abstract—**We develop load balancing algorithms for WDM-based packet networks in which the average traffic between nodes is dynamically changing. In WDM-based packet networks, routers are connected to each other using wavelengths (lightpaths) to form a logical network topology. This logical topology may be reconfigured by rearranging the lightpaths connecting the routers. The goal of our load balancing algorithms is to minimize network delay by reconfiguring the logical topology. Since delay becomes unbounded as the load approaches the link capacity, delay is usually dominated by the most heavily loaded link. Therefore, our algorithms attempt to minimize the maximum link load.**

Even when traffic is static, deriving the optimal logical topology for a given traffic pattern is known to be NP-complete. Previous work on reconfiguration proposed heuristic algorithms to determine the "best" logical topology for the given traffic pattern and migrated to that topology using a series of reconfiguration steps. However, when traffic patterns are changing rapidly, reconfiguring the full network with every change in the traffic may be extremely disruptive.

In this paper, we develop iterative reconfiguration algorithms for load balancing that track rapid changes in the traffic pattern. At each reconfiguration step, our algorithms make only a small change to the network topology, hence, minimizing the disruption to the network. We study the performance of our algorithms under several dynamic traffic scenarios and show that our algorithms perform near optimally.

Keywords—**Dynamic reconfiguration, load balancing, WDM networks.**

I. Introduction

WAVELENGTH Division Multiplexing (WDM) allows the enormous capacity of optical fiber to be utilized by transmitting multiple signals, distinguished by their wavelength, on a single fiber. Each wavelength (channel) operates at peak electronic speeds providing a capacity of 1 to 10 Gbps per channel. Many commercial systems that achieve 32 to 40 wavelengths per fiber have already been developed and systems employing nearly 100 wavelengths are forthcoming.

In metropolitan and wide area networks, most connections are multihop, i.e., most of the traffic is processed by intermediate electronic routers between the source and destination [1], [2]. Currently, WDM systems are still limited by costs of electronic components. Electronically processing each wavelength at each network node is prohibitively expensive as well as inefficient since much of the traffic traveling through a node may be destined for a downstream node. Optical Add/Drop Multiplexers (ADMs) and cross-connects may be used to allow individual wavelength signals to be either *dropped* to the electronic routers at each node or to pass through the node optically.

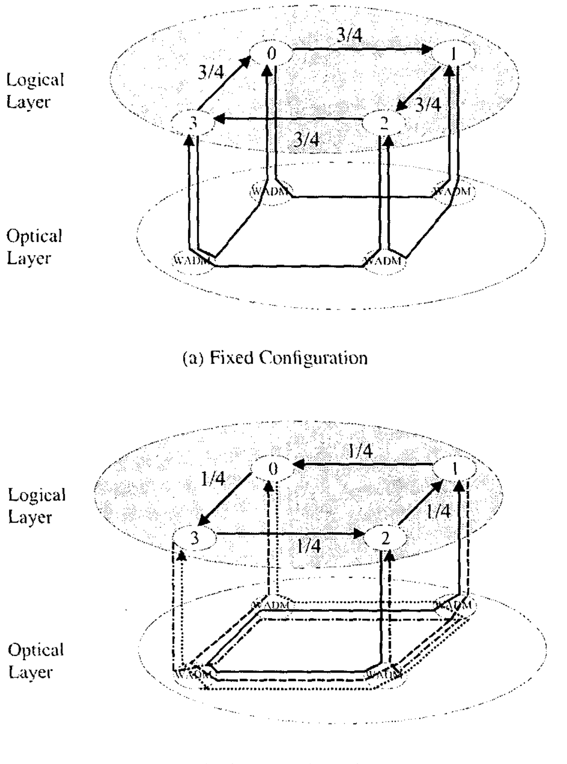
The passive or configurable optical nodes and their fiber connections constitute the *physical topology* of the network. The *logical topology* describes the lightpaths between the electronic routers and is determined by the configuration of the optical ADMs and transmitters and receivers on each node. Configurable components allow the logical topology of the network to be reconfigured. This capability can be used to reduce the traffic load on the electronic routers in accordance with the traffic pattern. Consider, for example, the traffic matrix shown in Figure 1, in which a four node network has 1/4 units of traffic between nodes i and $(i+3)$ mod 4. In Figure 2, we consider routing this traffic on a unidirectional ring physical topology with four nodes connected in the clockwise direction. In Figure 2(a), the logical topology is also configured in the clockwise direction. With this configuration, each logical link has a load of 3/4 units. However, by reconfiguring the logical topology to a counter-clockwise ring as shown in Figure 2(b), the logical link load can be reduced by a factor of 3. This reconfiguration reduces the amount of traffic that must be processed by each electronic router and hence the queuing delay.

In this work we develop reconfiguration algorithms to reduce the maximum link load in the network. We consider an N node network with an arbitrary but connected physical topology. Each node is assumed to have a small

This work was supported by DARPA under the Next Generation Internet (NGI) initiative. Opinions, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Air Force

$$S = \begin{matrix} 0 & 0 & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \end{matrix}$$

Fig. 1. Traffic matrix for a 4-node network that forms a ring pattern. Entry (i, j) denotes the traffic from node i to node j.



(a) Fixed Configuration

(b) Optimal Configuration

Fig. 2. Comparison of the load on the electronic routers under fixed and optimally reconfigured logical topology configurations. The load is based on the traffic matrix in Figure 1. The physical topology is assumed to be a unidirectional clockwise ring. The solid, dashed, dash-dot, and dotted lines in the optical layer represent the multiple wavelengths that are used to set up the lightpaths in the logical layer.

number, P, of transceiver ports. We assume that the number of wavelengths is unlimited and thus every virtual topology with P ports per node can be realized. Clearly, at most $W = PN$ wavelengths are needed to realize any possible logical topology. The exact number of wavelengths that are needed to implement a particular virtual topology depends on the physical topology of the network and can be much smaller than PN through wavelength reuse. In [3] the impact of restricting the number of available wavelengths is examined on networks with ring physical topologies.

The problem of determining the optimal logical topology in order to minimize the maximum link load consists of two subproblems: (1) the lightpath connectivity problem and (2) the traffic routing problem. In order to achieve the optimal topology, the two problems must be solved jointly. However, the topology design problem itself is NP-complete [4], thus typically the two problems are solved separately [5]. If each node is only equipped with a single transceiver port, $P = 1$, only a single path exists between each pair of nodes, thus there is no routing problem. In the case of multiple transceivers per node, flow deviation methods [6] can be used to find the optimal routing that minimizes the maximum link load for a given topology configuration. For simplicity, however, we focus on minimum hop routing. In addition to simplicity, minimum hop routing is attractive because it minimizes the total network load and is commonly used by network protocols.

As network traffic changes with time, the optimal logical topology varies as well. Although it is possible to implement any logical topology on our physical topology, changing the logical topology can be disruptive to the network since the traffic at each node must be buffered or re-routed while the topology is being reconfigured. With present day technology, transceivers and ADMs can be reconfigured in a few milliseconds (ms), therefore we expect the process of reconfiguration to take on the order of 10's of ms. At a gigabit per second rates, this delay corresponds to tens of megabits of traffic that must be re-routed or buffered at each node that is reconfigured. It is therefore important to reconfigure the network in a manner that limits the network disruption. Labourdette and Acampora [7] have suggested strategies to move from the current logical topology to the optimal logical configuration in small steps (branch exchange sequences) in order to reduce the impact on the network during reconfiguration. However, this approach results in a multi-step reconfiguration process that can take a long time and lead to topologies that are obsolete by the end of the process due to continually changing traffic patterns. Furthermore, intermediate reconfiguration steps may result in a temporarily disconnected network. For these reasons, it has been suggested that network reconfiguration should be utilized sparingly, perhaps only a few times a day. Rouskas and Ammar [8] have examined policies that reconfigure the network only when the benefits of reconfiguration outweigh the costs of reconfiguration. Tradeoffs between the costs and benefits of reconfiguration have also been addressed for single-hop broadcast Local Area Networks with slowly tunable receivers where reconfiguration is used to balance the load among the wavelengths [9], [10].

In this paper, we examine a multihop network reconfiguration strategy that makes a small change to the logical topology at regular intervals in order to reduce the network load. The small changes in the topology limit the disruption to the network allowing reconfiguration to be employed more often. Network reconfigurations at regular intervals allow the logical topology to track changes in traffic patterns. We show that our reconfiguration algorithm achieves performance improvements that are very close to optimal in the case of dynamic traffic. When traffic is static, the optimization steps, which are performed at regular intervals, converge to a locally optimal maximum load that is often very close to the global optimum.

II. Traffic Models

In this study, we assume that the traffic matrix specifying the traffic between each pair of nodes at a given time is known. We first describe static traffic models and then present a simple model to emulate time variation. For an N node network, let T be an $N \times N$ traffic matrix, where each entry $T_{i,j}$ represents the average rate of traffic from source node i to destination node j. Define $S = \frac{T}{\sum_{i,j} T_{i,j}}$ as a *normalized* traffic matrix, i.e., S is normalized to have total traffic $\sum_{i,j} S_{i,j} = 1$. Each entry $S_{i,j}$ represents the portion of the total traffic that is between nodes i and j. For load balancing, it is the relative traffic between pairs of nodes rather than the absolute traffic that is important.

The benefits provided by reconfiguration are inherently dependent on the traffic in the network. Consider, for example, a network in which each node has a single transceiver and that all logical topologies must be connected. If the traffic is all-to-all and uniform, i.e., $S_{i,j} = \frac{1}{N(N-1)}$ for all $i \neq j$, then the maximum link load is independent of the choice of logical topology and reconfiguration provides no benefits. If, however, the traffic pattern forms a ring (ring traffic is described in greater detail below), as shown in Figure 1, then the best topology, Figure 2 (b), results in a maximum link load of $\frac{1}{N}$ whereas the worst case topology, Figure 2 (a), results in a maximum link load of $\frac{N-1}{N}$. In this case, reconfiguration has the potential of reducing the maximum load by a factor of $N - 1$. More practical traffic models, such as those described below, lead to more limited reconfiguration gains.

We evaluate our algorithms utilizing three random traffic models. Similar traffic models were also used in [11], [5]. The resulting random matrices are normalized to have total traffic equal to 1 as described above.

1. *I.i.d.:* In this random traffic model, the traffic between each pair of nodes is independent and identically distributed (i.i.d.) with a uniform distribution between 0 and 1.

2. *Clustered:* In a traffic cluster, significant proportions of the traffic flow from a single source to multiple destinations or from multiple sources to a single destination. This type of traffic is representative of a file server model in which there are large flows from the file server to several users and also of a data collection model where there are large flows from several sites to a processor node. Alternatively, if the network nodes are aggregation points, clustered traffic is quite natural. In a clustered traffic matrix, the average traffic within a cluster is greater than the traffic between non-cluster connections by some factor. Multiple simultaneous clusters may coexist within a traffic matrix. A specific example of clustered traffic will be given in Section III-B.

3. *Ring:* Ring traffic is formed by selecting a random ordering of the network nodes $i_1, i_2, \ldots, i_N$, and generating uniform traffic between nodes i_j and $i_{(j+1) \bmod N}$. All other entries in the traffic matrix are zero. Although this traffic model is less realistic than the preceding models, it demonstrates the potential of reconfiguration.

We expect larger performance improvements from reconfiguration as the traffic becomes more structured. Under the i.i.d. traffic model, reconfiguration provides a small benefit, as the traffic between all source and destination pairs is uncorrelated. Larger improvements are expected for clustered traffic, especially as the cluster loading factor increases. For clustered and ring traffic, the traffic has a structure that can be exploited by an appropriate choice of the logical topology.

Let $S(t)$ denote the traffic matrix at time t. To model the dynamic nature of the traffic, we assume that two traffic matrices separated by time Δ are uncorrelated. The traffic matrix, $S(n\Delta)$ is independently generated for each $n \in \mathcal{Z}$ using the traffic models described above. We then assume that the traffic evolves linearly from traffic pattern $S((n-1)\Delta)$ to traffic pattern $S(n\Delta)$ in K steps. Furthermore, we assume that the traffic is constant in between linear interpolation steps, i.e., the traffic is constant for intervals of size $\delta = \Delta/K$. Formally,

$$S((n-1)\Delta + k\delta) = \\ S((n-1)\Delta) + \frac{k[S(n\Delta) - S((n-1)\Delta)]}{K}, \quad (1)$$

for $0 \leq k \leq K$ and

$$S((n-1)\Delta + k\delta + \tau) = S((n-1)\Delta + k\delta), \quad (2)$$

for $0 \leq \tau < \delta$. In this model, the traffic between each pair of nodes as a function of time is piecewise constant, taking K steps to move from the traffic load at time $(n-1)\Delta$ to the traffic load at time $n\Delta$.

The normalized traffic matrix S represents the shape of the traffic pattern. We assume that all links have equal capacity of C. As the link load approaches the link capacity, delay becomes unbounded. Therefore, the traffic that can be supported by the network is limited by the maximally loaded link. Let $l_{\max}(S, \theta)$ be the maximum link load under configuration θ and traffic pattern S. Then the link utilization on the maximally loaded link is $\frac{l_{\max}(S,\theta)}{C}$. If, for example, we require a maximum utilization of 90% in order to limit delay, the largest traffic matrix with shape S that can be supported under configuration θ is $\frac{.9C}{l_{\max}(S,\theta)} S$. Reconfiguring the logical topology and reducing the maximum link load increases the amount of traffic with shape S that can be supported or equivalently reduces queuing delay.

In the absence of reconfiguration, the logical topology is a fixed configuration, $\theta_{\rm fix}$. The optimal topology configuration, for a given traffic pattern, is the topology $\theta_{\rm opt}(S) = \arg\min_\theta l_{\max}(S.\theta)$ that minimizes the maximum link load. To characterize the benefits of reconfiguration, we calculate the reduction in maximum link load $\gamma_{\rm opt}(S) = \frac{l_{\max}(S,\theta_{\rm fix}) - l_{\max}(S,\theta_{\rm opt})}{l_{\max}(S,\theta_{\rm fix})}$ achieved by optimally reconfiguring the logical topology. We can similarly define the load reduction for alternative (suboptimal) reconfiguration algorithms.

III. Case I: Single Transceiver Per Node

We begin by considering reconfiguration for a network in which each node is equipped with a single transmitter and receiver. Maintaining a connected topology, which ensures that traffic between every source and destination pair can be continually supported, corresponds to requiring logical topologies that form unidirectional rings. In this case, the determination of the optimal topology, i.e., the logical topology that minimizes the maximum link load, is simplified by the fact that there is only a single route between each pair of nodes and thus the routing problem is eliminated and only the connectivity problem remains.

For an N node network, there are $(N-1)!$ possible logical ring topologies. The optimal topology may be determined through an exhaustive search, however, this approach quickly becomes impractical for large N. Since the connectivity design problem is NP-complete (reduces to Optimal Linear Arrangement problem) [4], [12], several heuristic approaches have been developed to provide "near-optimal" logical topologies with low computational complexity [11]. Most of these heuristics are designed to directly compute the "optimized" configuration for a static traffic matrix. In contrast, we develop gradient search algorithms for optimizing the logical topology in the presence of dynamically changing traffic. These approaches can also be used to obtain near-optimal logical topologies when traffic is static.

Since gradient search methods often lead to locally optimal rather than globally optimal solutions, prior approaches have included simulated annealing and gradient search algorithms employing multiple starting points [11], [13] to escape local minima. These approaches, however, are extremely computationally intensive and may not be useful when tracking dynamic traffic patterns.

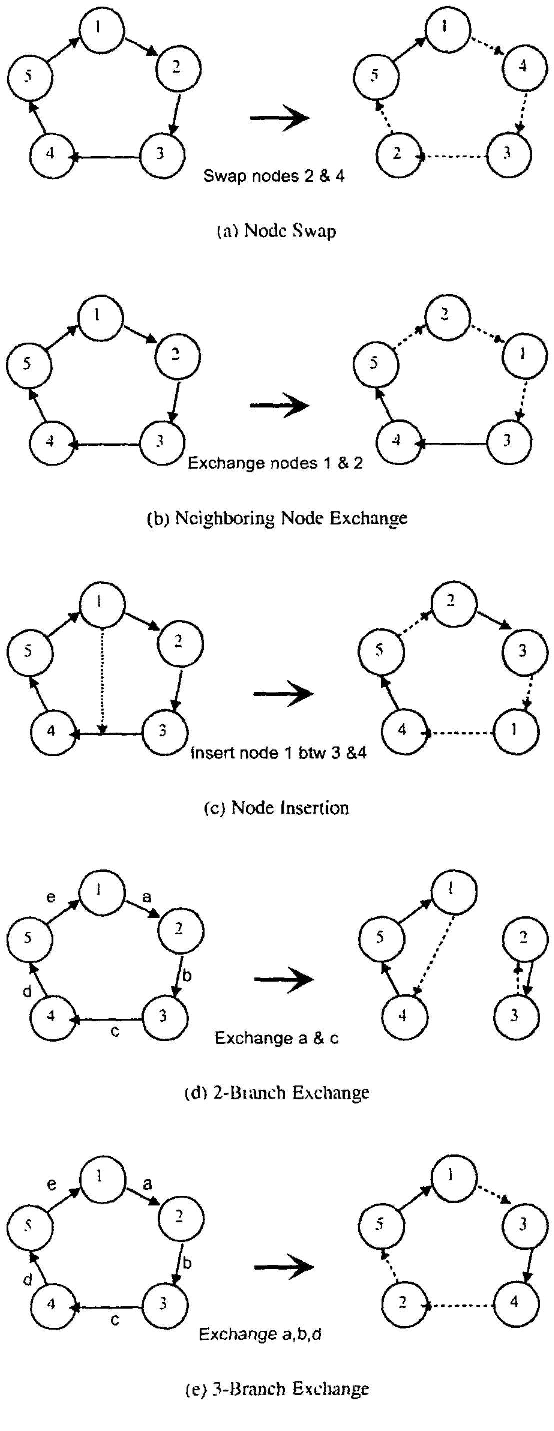
A. Local Exchanges

The algorithm we propose is an iterative local search algorithm which starts with a given topology and makes small "local" changes to the topology that reduce the load on the most heavily loaded link. Local search algorithms have been shown to produce good results for many combinatorial optimization algorithms [14].

Let Θ denote the set of all possible logical topologies and let $\theta \in \Theta$ be a particular logical topology. Then we can define a *neighborhood* as a mapping, for each logical topology $\theta \in \Theta$ to a set $\mathcal{N}(\theta)$ of logical topologies that are "close" in some sense to logical topology θ. At each iteration, a local search algorithm allows the logical topology θ_n to change to a logical topology within its neighborhood, $\theta_{n+1} \in \mathcal{N}(\theta_n)$. The crux of the local search algorithm is to determine a "good" exchange neighborhood. In our problem, we shall see that there is a very natural choice for the exchange neighborhood since our goals are to maximize the reduction in maximum flow while minimizing network disruption.

Shown in Figure 3 are several approaches that may be used to define neighborhoods for a ring logical topology. The first three approaches are based on moving network nodes. The first method, *node swap*, exchanges the locations of any two nodes in the ring. There are $\binom{N}{2}$ ways to select which nodes to swap and the resulting change can effect up to four links in the network. The second method, a *neighboring node exchange*, reverses the order of two neighboring nodes. There are N possible neighboring pairs that can be exchanged and each change disrupts three links. Alternatively, one might consider moving one node to a new location in the ring, i.e., a *node insertion*. This method yields a neighborhood of size $N(N-2)$ and also disrupts three links with each change. The next two methods view changing the network by moving network links rather than nodes. The first link method, the *2-branch exchange*, selects two links and exchanges their destinations. Unfortunately, a *2-branch exchange* in a unidirectional ring results in a disconnected logical topology. The final method we consider, a *3-branch exchange*, selects three links numbered 1, 2, and 3, in the order that they appear in the ring and connects the source of link 1 to the destination of link 2, the source of link 2 to the destination of link 3, and the source of link 3 to the destination of link 1. This 3-branch exchange leads to an exchange neighborhood of size $\binom{N}{3}$ where each network change disrupts three links. It can be shown that the 3-branch exchange always maintains ring connectivity. Furthermore, retaining ring connectivity requires reconfiguring a minimum of three links with each network change. Since the 3-branch exchange provides the maximum flexibility for modifying the logical topology ($\binom{N}{3}$ choices) while simultaneously minimizing the network disruption, we utilize the 3-branch exchange as our local exchange algorithm.

In Section III-B below, we examine the performance characteristics of the 3-branch exchange algorithm on a static traffic matrix. When the traffic is constant, multiple iterations of the 3-branch exchange algorithm will converge to a local minimum. An advantage of this approach under static traffic is that at each iteration, the network topology is improved while the disruption to the network is minimized. The method provides a natural way of migrating to a locally optimal topology. In Section III-C, we apply the algorithm to our dynamic traffic model. A single iteration of the algorithm, i.e., a single 3-branch exchange, is executed at each interval δ in Equation (1).



(a) Node Swap

(b) Neighboring Node Exchange

(c) Node Insertion

(d) 2-Branch Exchange

(e) 3-Branch Exchange

Fig. 3. Possible local exchange approaches used to define neighborhoods for a ring topology

B. Static Traffic Performance Results

We evaluate the performance of our algorithm on a network with $N = 10$ nodes[1]. Random traffic matrices are generated according to the traffic models described in Section II. The clustered traffic matrix is generated by taking a random i.i.d. traffic matrix, where each entry is i.i.d. and uniformly distributed, and weighting the traffic between nodes in a cluster by a cluster loading factor β. The cluster nodes are selected randomly. In the simulations of a 10 node network, we assume two non-overlapping clusters of 5 nodes. In the first cluster, a single source node is sending high volume of traffic to 4 destinations and in the second cluster, a single destination node is receiving high volume traffic from 4 sources. Note that a cluster of size 10 corresponds to a node sending (receiving) large amounts of traffic to (from) all other nodes. In this case, the traffic for the egress (ingress) node is similar to uniform all-to-all traffic and thus all connectivity patterns for the egress (ingress) node result in comparable performance. Clusters of size 2 are used in the ring traffic model. The cluster loading factor β is assumed to be 20. Larger cluster loading factors result in greater reconfiguration benefits whereas a cluster loading factor of $\beta = 1$ is equivalent to i.i.d. traffic.

Since traffic is static, reconfiguration at regular intervals permits multiple local exchanges to be executed with one exchange at each iteration. For a given static traffic matrix, the steepest descent algorithm proceeds as follows:

Step 1: Start with arbitrary initial topology, i.e., ring ordering.

Step 2: Search all $\binom{N}{3}$ 3-branch exchanges to determine which 3-branch exchange maximally reduces the maximum link load.

Step 3: If the best 3-branch exchange reduces the maximum link load, implement the change and goto Step 2, otherwise the algorithm has converged to a local optimum.

For networks of size $N \geq 5$, there exist traffic matrices and initial configurations for which the 3-branch exchange steepest descent algorithm converges to a point that is not globally optimal. For small values of N ($N = 10$ included), we can determine the optimal configuration by computing the maximum flow under all possible configurations via exhaustive search. We can then compare the performance of the 3-branch exchange and optimal reconfiguration strategies in terms of the reduction in maximum link load relative to a fixed configuration. Since the traffic is randomly generated, the fixed logical topology, θ_{fix}, may be any arbitrary ring ordering, without loss of generality. Define $\theta_{3be}(S)$ as the optimum configuration determined by the 3-branch exchange steepest descent algorithm. Then the reduction in maximum link

[1] Typical ring networks used in metropolitan and wide area networks have approximately 10 nodes, and are limited to a maximum of 16 nodes

load achieved by using the 3-branch exchange algorithm is $\gamma_{3be}(S_i) = \frac{l_{\max}(S_i,\theta_{\mathrm{fix}}) - l_{\max}(S_i,\theta_{3be})}{l_{\max}(S_i,\theta_{\mathrm{fix}})}$.

Figure 4 shows the maximum link load reduction as a function of the ith random traffic pattern. The 3-branch exchange steepest descent algorithm performance is compared to optimal reconfiguration for the i.i.d., clustered, and ring traffic models. Table I shows the average maximum link load reduction $\bar{\gamma}$ averaged over 1000 random traffic matrices for both the optimal and 3-branch exchange reconfiguration algorithms. For both the i.i.d. and clustered traffic models, the 3-branch exchange algorithm performs very close to optimal. As expected, reconfiguration results in larger performance improvements in the case of clustered traffic. For ring traffic, both the 3-branch exchange and optimal reconfiguration strategies produce very significant improvements. This traffic model highlights the limitations of the 3-branch exchange algorithm relative to optimal reconfiguration. However, this traffic model is not very realistic, and even in this case the 3-branch exchange algorithm provides significant improvements over a fixed topology. One method of escaping local minima in a gradient descent algorithm is to utilize several initial starting configurations [11], [14]. The results for ring traffic indicate that when network traffic is static, it may be advantageous to compute the globally optimal configuration to determine if the network should be moved from the locally optimal solution.

TABLE I

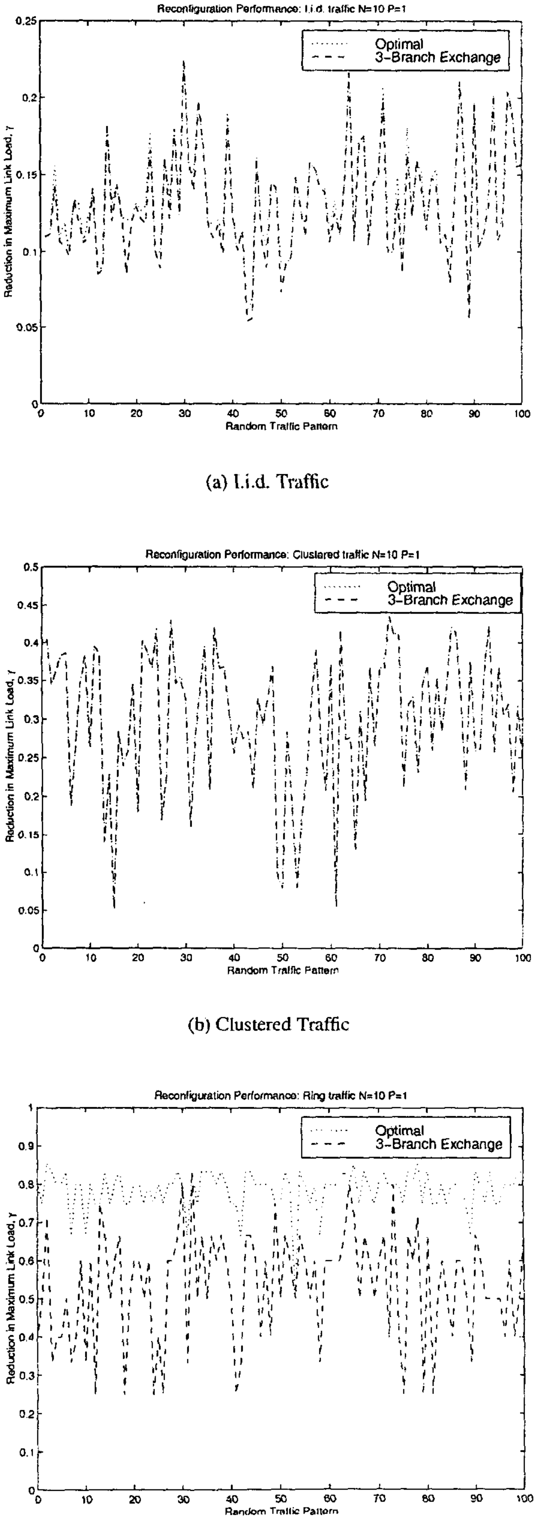
AVERAGE REDUCTION IN MAXIMUM LINK LOAD

Traffic model	3-branch exchange	Optimal
I.i.d.	0.13	0.14
Clustered	0.29	0.29
Ring	0.61	0.80

The convergence properties of the 3-branch exchange algorithm are reported in Table II. Through simulations we determine the percentage of time the algorithm converges to the optimal solution. We also compute the average and maximum number of iterations the algorithm takes to converge. Although the 3-branch exchange algorithm does not always converge to the global optimal, simulations show that for a 10 node network, 98% of the local minima are within 2% of the global minimum under i.i.d. traffic and 99% of the local minima are within 1.5% of the global minimum under clustered traffic.

C. Dynamic Traffic Performance Results

If significant changes in the traffic patterns are occurring very rapidly, frequent full-network reconfigurations to the optimal topology may excessively disrupt traffic while infrequent reconfigurations may result in out-dated configuration patterns and suboptimal loading conditions. A Dynamic Single-Step Optimization (DSSO) algorithm that



(a) I.i.d. Traffic

(b) Clustered Traffic

(c) Ring Traffic

Fig. 4. Reduction in maximum link load achieved via reconfiguration using a 3-branch exchange steepest descent algorithm on a network with $N = 10$ nodes and $P = 1$ port per node.

TABLE II

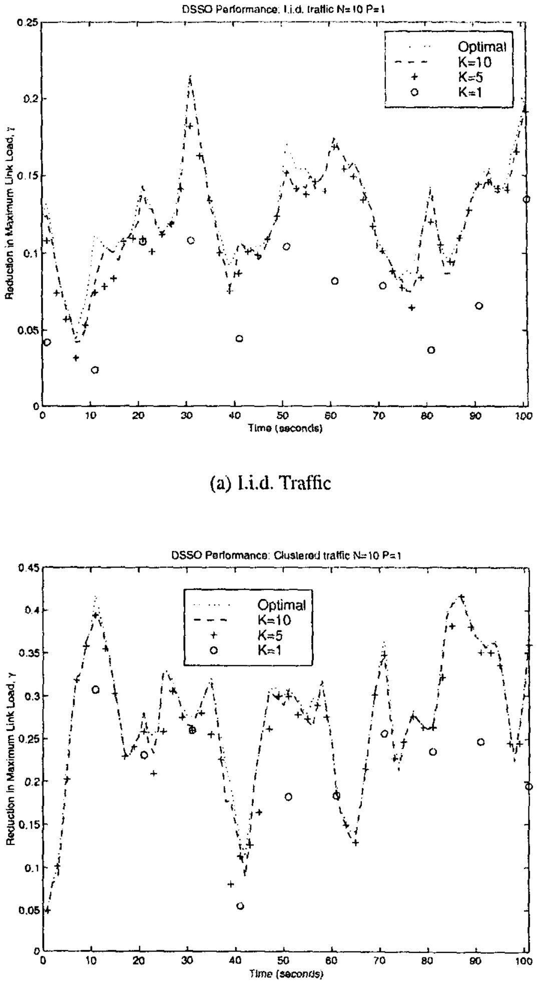
CONVERGENCE PROPERTIES OF 3-BRANCH EXCHANGE ALGORITHM

	Traffic model		
	I.i.d.	Clustered	Ring
% Convergence to optimal	53.5	66.2	10.4
Avg. iters to converge	4.7	4.9	2.4
Max. iters to converge	10	8	4

implements a single 3-branch exchange at regular intervals provides a natural method of reducing the network disruption while tracking the optimal configuration pattern.

To evaluate the performance of this algorithm, we utilize the dynamic traffic model described in Section II where the traffic matrix evolves from one independent traffic pattern to another in K steps. At each of the K steps, we implement the 3-branch exchange that maximally reduces the maximum link flow. Smaller values of K correspond to sampling the time-varying traffic patterns more coarsely which forces each 3-branch exchange iteration to reconcile larger traffic changes. Our results for static traffic patterns showed that on average 4.8 iterations were needed to converge to a local optimum for a randomly selected traffic matrix. Therefore we expect that if $K > 5$, the 3-branch exchange should be able to closely track the optimal configuration. When $K = 1$, the traffic pattern is uncorrelated at each time step and we expect the performance of a single optimization step to be more limited. By varying the number of steps between independent traffic conditions, K, we measure the ability of the DSSO algorithm to track random traffic fluctuations.

Consider a scenario where the traffic matrix evolves from one independent traffic matrix to another in 10 seconds. A value of $K = 10$ corresponds to sampling the time-varying traffic matrix once each second. At each sample time, a 3-branch exchange is implemented. Our traffic model assumes that the traffic is constant between sampling intervals, thus, for $K = 10$, the traffic matrix changes once each second. For $K = 5$ the traffic matrix changes once every two seconds, but each change is twice as large. In Figure 5, the reduction in maximum link load achieved by the optimal configuration strategy $\gamma_{\text{opt}}(S_i)$ and by the DSSO algorithm $\gamma_{DSSO}(S_i)$ are shown for several values of K. When $K = 10$, the DSSO algorithm closely tracks the optimal configuration. Figure 6 shows the time average reduction in maximum link load as a function of K. The DSSO algorithm provides significant reductions in maximum link load over fixed configuration systems even when $K = 1$ and a single 3-branch exchange must accommodate a complete change in the traffic pattern.



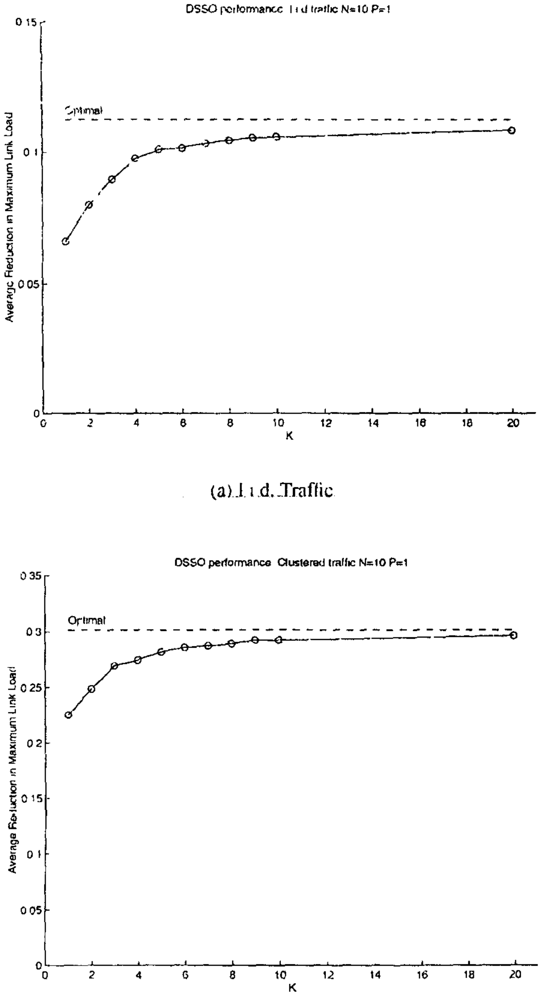
(a) I.i.d. Traffic

(b) Clustered Traffic

Fig. 5. Reduction in maximum link load achieved with DSSO algorithm and optimal reconfiguration strategies under dynamic traffic on a network with $N = 10$ nodes and $P = 1$ port per node.

IV. CASE II: MULTIPLE TRANSCEIVERS PER NODE

In a network consisting of nodes equipped with multiple transceiver ports, multiple (multihop) paths exist between each source and destination pair. Therefore the optimal logical topology for load balancing is a function of both the lightpath connectivity and traffic routing. An optimal solution requires jointly solving the connectivity and routing problems. However, since the problem is difficult to solve jointly, most approaches, including ours, separate the two problems. We focus on solving the connectivity problem, which for a network with N nodes and P transceivers per node consists of on the order of $(N!)^P$ possible logical topologies. Although optimal routing to minimize maximum flow can be achieved through flow de-



(a) I.i.d. Traffic

(b) Clustered Traffic

Fig. 6 Time average reduction in maximum link load resulting from DSSO reconfiguration as a function of K the number of steps between independent random traffic patterns. Results are for a network with $N = 10$ nodes and $P = 1$ port per node.

viation methods, the resulting routing protocol is computationally intensive. For simplicity we assume minimum hop routing which minimizes the total network load and is often used in packet routing protocols. Clearly, however, our connectivity algorithms could be used in conjunction with optimal routing methods providing performance improvements for both the reconfigurable and fixed topologies. We expect the benefits of reconfigurable topologies over fixed topologies to be similar for both optimal and minimum hop routing.

With two ports per node, there are many possible fixed topologies that can be established. We compare the reconfigurable topologies to a fixed logical topology $\theta_{\rm fix}$ of a bidirectional ring. When compared to alternative fixed logical topologies, such as the perfect shuffle, reconfiguration provided similar improvements.

The size of the connectivity problem prohibits an exhaustive search to determine the optimal logical topology. Therefore we compare our algorithm's performance to lower bounds on the minimum maximum flow. These bounds are similar to those derived in [5]. In a network with N nodes and P ports there are PN one hop connections, P^2N two hop connections, etc. Using linear programming techniques, we can determine the PN connections that carry the largest amount of traffic in a single hop, given the port restrictions. Thus a lower bound on the total carried traffic, τ, may be computed by assuming that the corresponding PN traffic elements are carried in one hop, the next P^2N largest traffic elements are carried in two hops, etc. Ideally, this total traffic could be divided evenly among the PN links, yielding the lower bound:

$$l_{\max} \geq \frac{\tau}{PN}. \tag{3}$$

Next, we note that minimum hop routing does not bifurcate the traffic. Thus, the maximum traffic element, $\max_{i,j} S_{i,j}$ must traverse a single link. Furthermore, the total traffic leaving or entering a single node is at best divided evenly among P links, producing a second lower bound:

$$l_{\max} \geq \max(\max_{i,j} S_{i,j}, \frac{\max_i \sum_j S_{i,j}}{P}, \frac{\max_j \sum_i S_{i,j}}{P}). \tag{4}$$

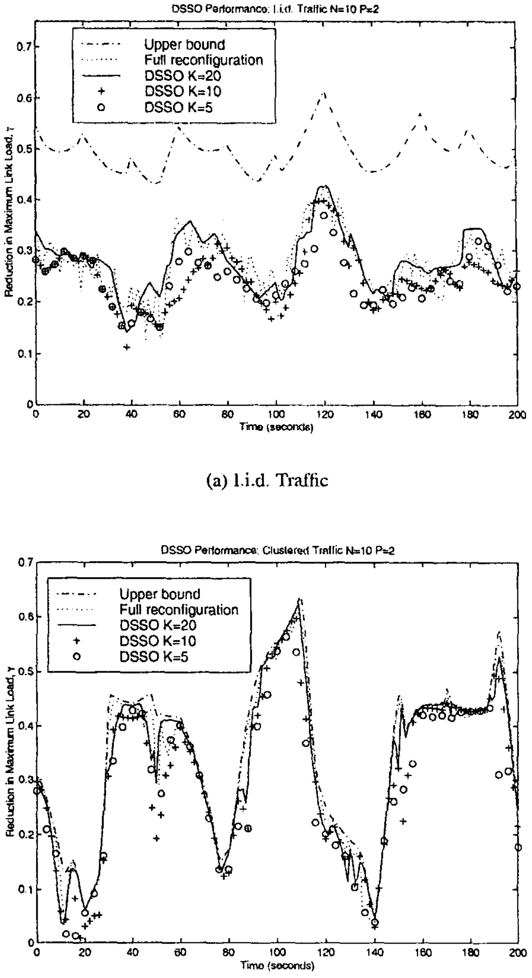
The lower bounds on maximum flow, denoted LB, are used to calculate upper bounds on the maximum reduction in link load, $\gamma_{\rm UB}(S) = \frac{l_{\max}(S,\theta_{\rm fix}) - LB(S)}{l_{\max}(S,\theta_{\rm fix})}$.

With multiple transceivers per node, the minimum network change that retains connectivity is a 2-branch exchange. However, not all 2-branch exchanges maintain connectivity. The following algorithm for selecting 2-branch exchanges is utilized to ensure network connectivity. First the initial configuration is assumed to be connected. Next only vertex disjoint branches, lightpaths which do not share a common vertex, are exchangeable. This ensures that all ports are always utilized, since exchanging non vertex disjoint branches may result in a connection between a transmitter and receiver on the same node. Finally, alternate routes for lightpaths that are removed are verified in the new configuration using, for example, a shortest path routing algorithm. Branch exchange methods have been applied previously to a number of topological problems including topology design for static traffic conditions [15], [16]. In this work, we utilize branch exchange methods in a dynamic algorithm and evaluate the algorithm's capabilities under time-varying traffic.

The 2-branch exchange is used to implement a Dynamic Single Step Optimization (DSSO) algorithm for dynamic traffic. The dynamic traffic matrix, as described in Section II, evolves in K steps from one independent traffic matrix to another. At each step, the 2-branch exchange that maximally reduces maximum flow while retaining network connectivity is implemented. For clustered traffic, each independent matrix is constructed with two randomly located clusters of size 5 and with a loading factor of $\beta = 60$. A larger value of β is utilized for the multiple port per node case since an exhaustive search for the optimal topology is too computationally intensive and the upper bound becomes tighter as the traffic within the clusters becomes more dominant. Algorithm performance is examined for a network with 10 nodes and 2 transceiver ports per node.

In Figure 7, we show the reduction in maximum link load achieved by the DSSO algorithm, γ_{DSSO} for several values of K and compare this to the upper bound on maximum load reduction γ_{UB} as a function of time. We also illustrate the performance of the lightpath connectivity algorithm proposed by Labourdette and Acampora [5] which executes a full network reconfiguration at each step. The algorithm in [5] uses a linear program to determine the lightpath connectivity pattern that carries the largest amount of single hop traffic and improves the connectivity pattern using a sequence of 2-branch exchanges. We implemented the lightpath connectivity algorithm of [5] and show its performance in Figures 7 and 8, where it is referred to as 'Full reconfiguration'.

We assume in Figure 7 that the traffic moves from one independent traffic matrix to another in 20 seconds. Thus, the DSSO algorithm implements a 2-branch exchange once every $20/K$ seconds. The DSSO algorithm provides a significant reduction in maximum link load over fixed configuration systems under both i.i.d. and clustered traffic models. Note that in the case of clustered traffic and $K = 20$, the DSSO algorithm performance approaches the upper bound, whereas for i.i.d. traffic the upper bound predicts significantly larger reductions. This effect is due to the tightness of the upper bound. For clustered traffic, the second lower bound provides a better approximation to the maximum achievable link load reduction since the maximum link load is dominated by the heaviest traffic flows and most congested nodes. For i.i.d. traffic, the bounds are not as tight. Comparisons of the upper bounds to optimal load reduction in the case of 1 port per node verify that the bounds are not tight, and are extremely optimistic for the case of i.i.d. traffic. For both i.i.d. and clustered traffic, we find that the performance of the DSSO algorithm, which executes a single 2-branch exchange at each iteration, is very close to the performance of the Labourdette and Acampora algorithm [5] which requires a full network reconfiguration at each step. Since both the DSSO algorithms and the full network reconfiguration algorithms are suboptimal configuration strategies, we see that the DSSO algorithm actually performs better than the full network reconfiguration algorithm in many cases. Figure 8 illustrates the time average reduction in maximum link load achieved by the DSSO algorithm as a function of K.



(a) I.i.d. Traffic

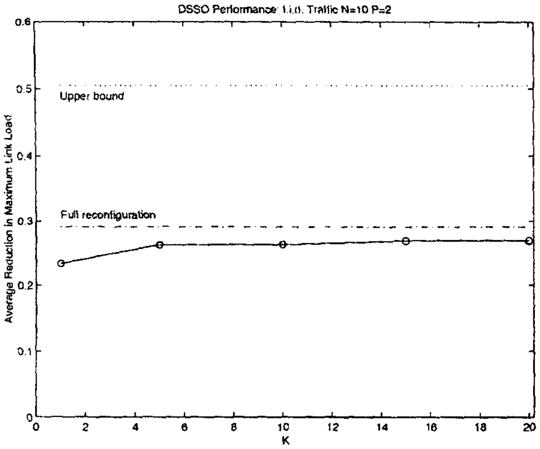
(b) Clustered Traffic

Fig. 7. Reduction in maximum link load achieved with DSSO algorithm under dynamic traffic conditions on a network with $N = 10$ nodes and $P = 2$ ports per node.

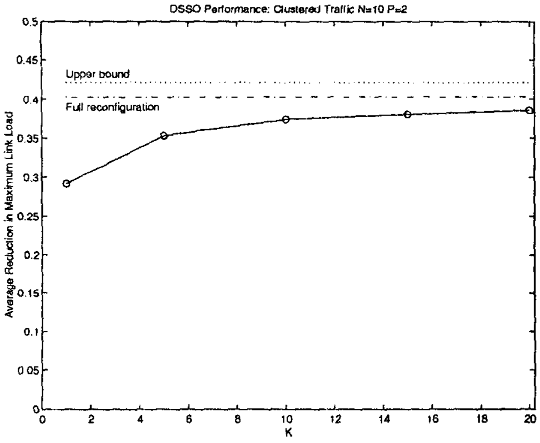
V. Conclusions

We developed and analyzed a reconfiguration strategy where small local changes to the network are applied at regular intervals. This strategy minimizes the network disruption at each iteration while allowing the logical topology to track changes in traffic conditions. This reconfiguration approach was shown to provide significant and near optimal reduction in maximum link load.

In situations where the traffic pattern is static or extremely slowly varying, the gradient descent algorithms



(a) I.i.d. Traffic



(b) Clustered Traffic

Fig. 8. Time average maximum link load reduction resulting from DSSO reconfiguration as a function of K, the number of steps between independent random traffic patterns. Results are for a network with $N = 10$ nodes and $P = 2$ ports per node.

may settle at a logical topology that is locally but not globally optimal. In this case, it may be useful to augment our approach with infrequent full network reconfigurations to the globally optimal topology. A method of effectively combining the two approaches is an area for future work.

In this work we assume that the number of wavelengths is unlimited and thus every virtual topology with P ports per node can be realized. Clearly, at most $W = PN$ wavelengths are needed to realize any possible logical topology. In [3] we examine the restrictions imposed by constraining the number of available wavelengths. We show that significant reductions in network load are achievable even when the number of available wavelengths is much smaller than PN.

Although the proposed reconfiguration algorithm limits the number of nodes reconfigured at each iteration, the process of reconfiguration may still result in network disruption. Clearly when network tuning and switching delays are small relative to the time between reconfigurations, this disruption will have minimal impact on network performance. It would be interesting to study the impact of the reconfiguration process when tuning delays are more significant.

Acknowledgments

The authors would like to thank Dr. Kimo Tam, Dr. Steve Finn, and Dr. Philip Lin for helpful discussions and suggestions.

References

[1] A. S. Acampora, "A multichannel multihop local lightwave network," in *GLOBECOM*, 1987, vol. 3, pp. 1459–1467.

[2] B. Mukherjee, "Wdm-based local lightwave networks part ii: Multihop systems," *IEEE Network*, pp. 20–32, July 1992.

[3] A. Narula-Tam and E. Modiano, "Dynamic reconfiguration in wdm packet networks with wavelength limitations," to appear in OFC, 2000.

[4] D. Bienstock and O. Gunluk, "Computational experience with a difficult mixed-integer multicommodity flow problem," *Mathematical Programming*, vol. 68, pp. 213–237, 1995.

[5] J. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Transactions on Communications*, vol. 39, no. 8, pp. 1223–1230, Aug. 1991.

[6] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: An approach to store-and-forward communication network design," *Networks*, vol. 3, pp. 97–133, 1973.

[7] J-F. P. Labourdette, G. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration of lightwave networks," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2822–2832, Oct. 1994.

[8] G. N. Rouskas and M. H. Ammar, "Dynamic reconfiguration in multihop wdm networks," *Journal of High Speed Networks*, vol. 4, no. 3, pp. 221–238, June 1995.

[9] I. Baldine and G. N. Rouskas, "Dynamic load balancing in broadcast wdm networks with tuning latenies," in *IEEE Infocom*, 1998, vol. 1, pp. 78–85.

[10] I. Baldine and G. N. Rouskas, "Dynamic reconfiguration policies for wdm networks," in *IEEE Infocom*, 1999, vol. 1, pp. 313–320.

[11] S. Banerjee and B. Mukherjee, "The photonic ring: Algorithms for optimized node arrangements," *Fiber and Integrated Optics*, vol. 12, no. 2, pp. 133–171, 1993.

[12] D. Bienstock, "Private communications," 1999.

[13] J. A. Bannister, L. Fratta, and M. Gerla, "Optimal topologies for the wavelength-division optical network," in *EFOC/LAN 90. The Eighth European Fibre Optic Communications and Local Area Networks Exposition. LAN Proceedings*, 1990, pp. 53–57.

[14] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice Hall-Inc., New Jersey, 1982.

[15] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks," *IEEE Transactions on Communications*, vol. COM-25, no. 1, pp. 48–60, Jan. 1977.

[16] H. Frank, I. T. Frisch, and W. Chou, "Topological considerations in the design of the arpa computer network," in *Conf. Rec., 1972 Spring Joint Comput. Conf., AFIPS Conf. Proc.*, 1972, vol. 40, pp. 255–270.