Dynamic Load Balancing in WDM Packet Networks With and Without Wavelength Constraints

Aradhana Narula-Tam and Eytan Modiano, *Senior Member, IEEE*

***Abstract*—We develop load balancing algorithms for WDM-based packet networks where 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. The logical topology may be reconfigured by rearranging the lightpaths connecting the routers. Our algorithms reconfigure the logical topology to minimize the maximum link load.**

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. We further show that these large reconfiguration gains are achievable in systems with a limited number of wavelengths.

***Index Terms*—Ring networks, topology reconfiguration, virtual topology design, wavelength division multiplexing (WDM).**

I. Introduction

WAVELENGTH division multiplexing (WDM) allows the enormous optical capacity of 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 1–10 Gb/s per channel. Many commercial systems that achieve 32–40 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 still 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.

In this work we develop reconfiguration algorithms to reduce the maximum link load in the network. We begin by considering an N node network with an arbitrary but connected physical topology. Each node is assumed to have a small number, P, of transceiver ports. We initially 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. We then examine the restrictions imposed by constraining the number of available wavelengths. With a limited number of wavelengths, multiple lightpaths must be routed on a single wavelength. To analyze the impact of wavelength limitations, we consider a wavelength routed network with ring physical topology. We show that significant reductions in network load are achievable even when the number of available wavelengths is much smaller than PN.

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 [3], thus typically the two problems are solved separately [4]. 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 [5] 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 rerouted while the topology is being reconfigured. With present-day technology, transceivers and ADMs can be reconfigured in a few milliseconds (ms),

Manuscript received October 15, 1999; revised May 15, 2000. This work was supported by the Defense Research Projects Agency (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.

A. Narula-Tam is with the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Lexington, MA 02420 USA.

E. Modiano is with MIT Laboratory for Information and Decision Systems, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0733-8716(00)09007-7.

therefore we expect the process of reconfiguration to take on the order of tens of ms. At a gigabit per second rate, this delay corresponds to tens of megabits of traffic that must be rerouted 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 [6] 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 multistep 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 [7] 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 [8], [9].

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 $S = (T/\Sigma_{i,j}\, T_{i,j})$ as a *normalized* traffic matrix, i.e., S is normalized to have total traffic $\Sigma_{i,j}\, S_{i,j} = 1$. 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.

We evaluate our algorithms utilizing two random traffic models—*i.i.d.* and *clustered*. Similar traffic models were also used in [10] and [4]. The resulting random matrices are normalized to have total traffic equal to 1 as described above. In the i.i.d. 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. In the clustered traffic model, the average traffic within clusters is greater than the traffic between noncluster connections by some factor. 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. If the network nodes are aggregation points, clustered traffic is also quite natural. Multiple simultaneous clusters may coexist in a traffic matrix. A specific example of clustered traffic will be given in Section III-B.

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 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 $\delta = \Delta/K$.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 $n\Delta$.

Let $l_{\max}(S, \theta)$ be the maximum link load under configuration θ and traffic pattern S. In the absence of reconfiguration, the logical topology is a fixed configuration, θ_{fix}. The optimal topology configuration, for a given traffic pattern, is the topology $\theta_{\text{opt}}(S) = \arg\ \min_\theta l_{\max}(S, \theta)$ that minimizes the maximum link load. To characterize the benefits of reconfiguration, we calculate the reduction in maximum link load

$$\gamma_{\text{opt}}(S) = \frac{l_{\max}(S, \theta_{\text{fix}}) - l_{\max}(S, \theta_{\text{opt}})}{l_{\max}(S, \theta_{\text{fix}})}$$

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.

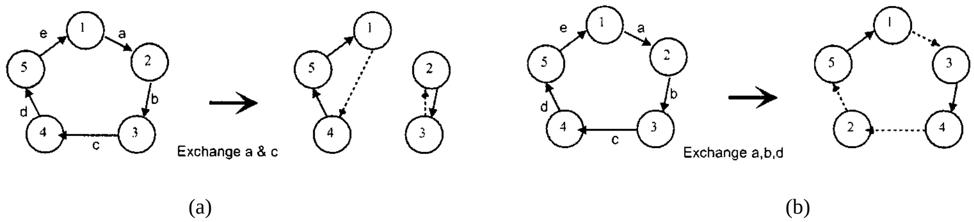


Fig. 1. Local exchange approaches applied to a ring topology. (a) 2-branch exchange. (b) 3-branch exchange.

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 the optimal linear arrangement problem) [3], [11], several heuristic approaches have been developed to provide "near-optimal" logical topologies with low computational complexity [10]. 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 [10], [12] 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 [13]. 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.

There are several approaches that may be used to define neighborhoods for a ring logical topology [14]. In this work, we utilize the two methods shown in Fig. 1. The first method, *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. Retaining ring connectivity requires reconfiguring a minimum of three links with each network change. 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 yields $\binom{N}{3}$ possible changes to the network topology, where each network change disrupts three links. It can be shown that the 3-branch exchange always maintains ring connectivity. Furthermore, the 3-branch exchange provides the maximum flexibility for modifying the logical topology [$\binom{N}{3}$ choices] while simultaneously minimizing network disruption [14].

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.

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 $N = 10$ node network, we assume two nonoverlapping clusters of 5 nodes. In the first cluster, a single source node is sending high volume of traffic to 4 destinations. 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. 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.

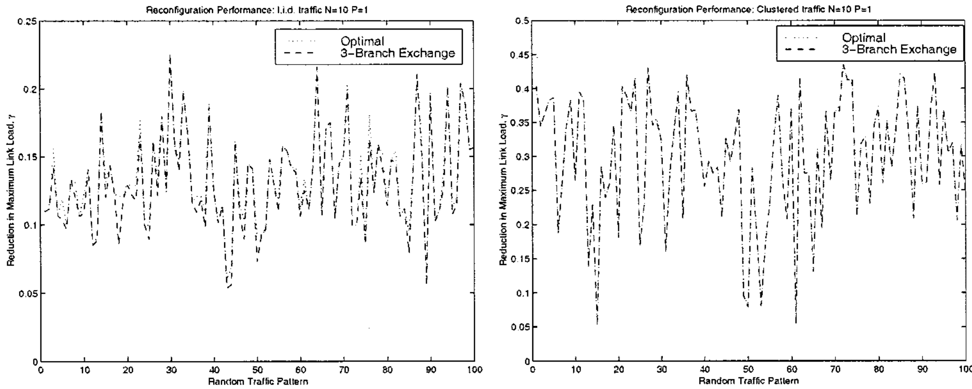
Since traffic is static, reconfiguration at regular intervals permits multiple local exchanges to be executed with one exchange at each iteration. For a given static traffic matrix, the steepest descent algorithm proceeds as follows.

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 go to Step 2.

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



(a) (b)

Fig. 2. 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. (a) I.i.d. traffic. (b) Clustered traffic.

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 load achieved by using the 3-branch exchange algorithm is

$$\gamma_{3be}(S_i) = \frac{l_{\max}(S_i, \theta_{\text{fix}}) - l_{\max}(S_i, \theta_{3be})}{l_{\max}(S_i, \theta_{\text{fix}})}.$$

Fig. 2 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. and clustered traffic models. The average maximum link load reduction $\overline{\gamma}$ averaged over 1000 random traffic matrices for the optimal and 3-branch exchange reconfiguration algorithms is $\overline{\gamma}_{3be} = 0.13$ and $\overline{\gamma}_{\text{opt}} = 0.14$, respectively, for i.i.d. traffic and $\overline{\gamma}_{3be} = 0.29$ and $\overline{\gamma}_{\text{opt}} = 0.29$ for clustered traffic. For both 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.

The convergence properties of the 3-branch exchange algorithm are reported in Table I. 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.

TABLE I
CONVERGENCE PROPERTIES OF 3-BRANCH EXCHANGE ALGORITHM

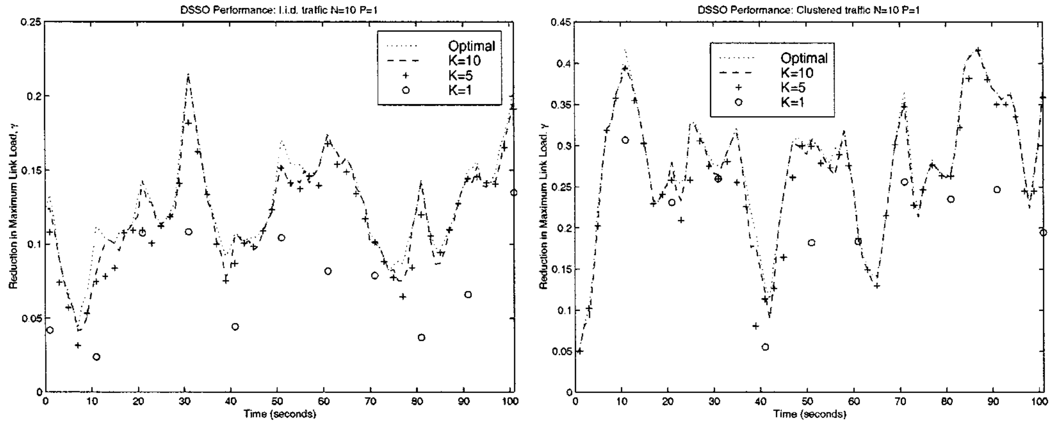
	Traffic model	
	I.i.d.	Clustered
% Convergence to optimal	53.5	66.2
Avg. number iterations to converge	4.7	4.9
Max. number iterations to converge	10	8

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 outdated configuration patterns and suboptimal loading conditions. A dynamic single-step optimization (DSSO) algorithm that 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. 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 Fig. 3, the reduction in maximum link load achieved



(a) (b)

Fig. 3. 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. (a) I.i.d. traffic. (b) Clustered traffic.

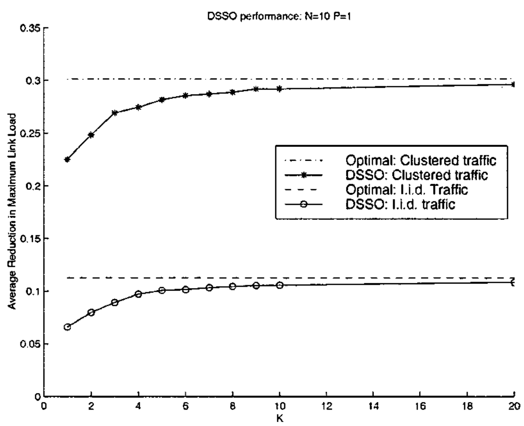


Fig. 4. Time average reduction in maximum link load resulting from DSSO reconfiguration as a function of K the number of steps between independent random traffic patterns. Results are for a network with $N = 10$ nodes and $P = 1$ port per node.

by the optimal configuration strategy $\gamma_{\text{opt}}(S_i)$ and by the DSSO algorithm $\gamma_{\text{DSSO}}(S_i)$ are shown for several values of K. When $K = 10$, the DSSO algorithm closely tracks the optimal configuration. Fig. 4 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 compensate a complete change in the traffic pattern.

IV. Case II: Multiple Transceivers Per Node

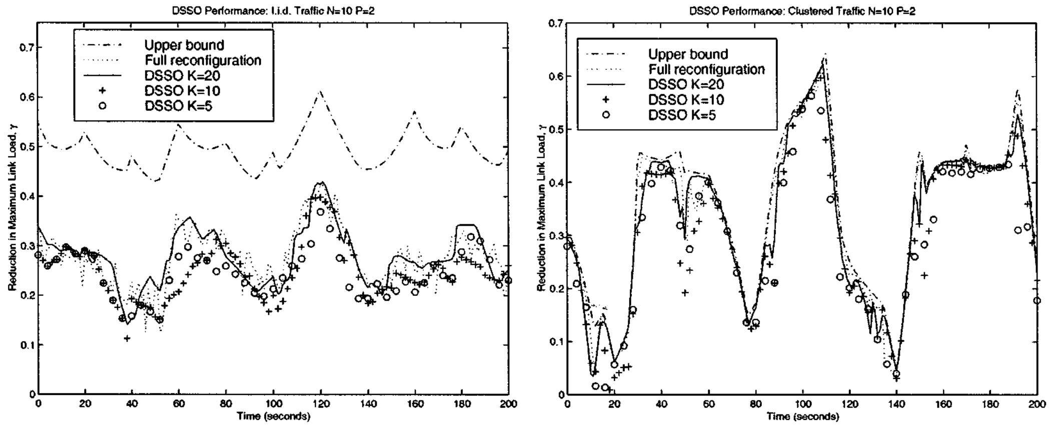
In a network consisting of nodes equipped with multiple transceiver ports, multiple (multihop) paths exist between each source and destination pair. Therefore, the optimal logical topology for load balancing is a function of both the lightpath connectivity and traffic routing. An optimal solution requires 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 $(N-1)!^P$ possible logical topologies. Although optimal routing to minimize maximum flow can be achieved through flow deviation 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 θ_{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 [4]. 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

$$l_{\max} \geq \frac{\tau}{PN}. \tag{1}$$

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,



(a) (b)

Fig. 5. 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. (a) I.i.d. traffic. (b) Clustered traffic.

producing a second lower bound

$$l_{\max} \geq \max\left(\max_{i,j} S_{i,j}, \frac{\max_i \sum_j S_{i,j}}{P}, \frac{\max_j \sum_i S_{i,j}}{P}\right). \quad (2)$$

The lower bounds on maximum flow, denoted LB, are used to calculate upper bounds on the maximum reduction in link load,

$$\gamma_{\mathrm{UB}}(S) = \frac{l_{\max}(S, \theta_{\mathrm{fix}}) - LB(S)}{l_{\max}(S, \theta_{\mathrm{fix}})}.$$

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 nonvertex 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

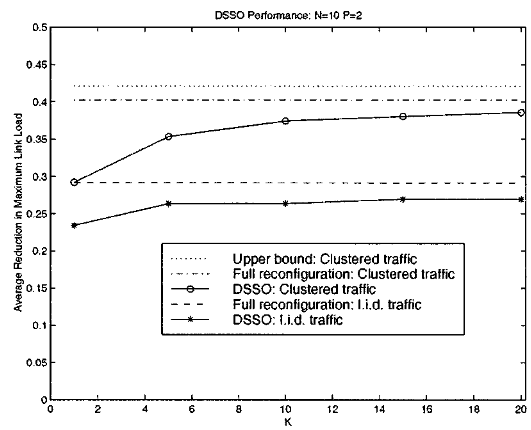


Fig. 6. Time average maximum link load reduction resulting from DSSO reconfiguration as a function of K, the number of steps between independent random traffic patterns. Results are for a network with $N = 10$ nodes and $P = 2$ ports per node.

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 $N = 10$ nodes and $P = 2$ transceiver ports per node.

In Fig. 5, 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 [4] which executes a full network reconfiguration at each change in the traffic matrix. The algorithm in [4] 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 [4] and show its performance in Figs. 5 and 6, where it is referred to as "full reconfiguration."

We assume in Fig. 5 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 $P = 1$ 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 [4] which executes a full network reconfiguration with every change in the traffic matrix. Since both the DSSO algorithms and the full network reconfiguration algorithms are locally optimal configuration strategies, we see that the DSSO algorithm actually performs better than the full network reconfiguration algorithm in many cases. Fig. 6 illustrates the time average reduction in maximum link load achieved by the DSSO algorithm as a function of K.

V. Impact of Wavelength Restrictions

Above we showed that reconfiguring the logical topology via the DSSO algorithm significantly reduces the maximum link load. We also showed that the DSSO algorithm closely tracks the optimal logical topology when the traffic is time varying. These results assumed that a sufficient number of wavelengths were available to implement all possible logical topologies. Each lightpath was guaranteed its own wavelength. We next examine the case where the number of wavelengths in the system is less than the number of lightpaths, $W < PN$. In this case, the set of logical topologies that can be established is a function of the physical topology.

With a ring physical topology, multiple lightpaths may be able to share a single wavelength. However, if the physical topology is a unidirectional ring, the worst case logical topology still requires PN wavelengths. The bidirectional ring offers better wavelength reuse properties. Therefore, for the remainder of this section we consider a bidirectional ring physical topology.

We assume each lightpath is routed using a fixed shortest path routing scheme. For lightpaths requiring $N/2$ hops (when N is even), lightpaths from odd nodes $1 \leq i \leq N/2$ to $i + N/2$ and from $i+N/2$ to i are routed clockwise. The remaining lightpaths are routed in the counterclockwise direction. Since the DSSO algorithms dynamically add and remove individual lightpaths from the logical topology, we use shortest path routing which minimizes resource utilization and is thus conducive to future logical topology changes.

In [17], the wavelength requirements for single port per node $(P = 1)$ networks using shortest path routing were determined. We consider the total number of wavelengths required as the number of wavelengths required in the clockwise direction plus

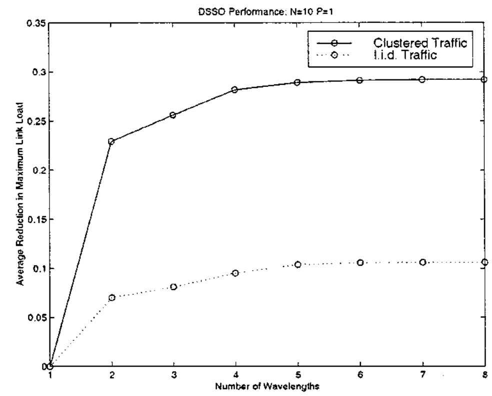


Fig. 7. The average reduction in maximum link load achieved by the DSSO reconfiguration algorithm is plotted as a function of the network wavelength restriction for dynamic traffic with $K = 10$. Results are for a network with $N = 10$ nodes and $P = 1$ port per node.

the number of wavelengths required in the counterclockwise direction. In order to implement all possible connected logical topologies, it was shown in [17] that a minimum of $N-2$ wavelengths are required.

Thus, in an $N = 10$ node system, a minimum of 8 wavelengths is required to implement all connected logical topologies. Further restrictions on the number of wavelengths imply that not all logical topologies are feasible. With a wavelength constraint, the DSSO algorithm is restricted to branch exchanges that do not require more than the appropriated number of wavelengths. Fig. 7 shows, for an $N = 10$ node network, the average reduction in maximum link load provided by the DSSO algorithm as a function of the number of available wavelengths. Note that the number of wavelengths required to achieve a large fraction of the possible reduction is quite small. This result may be explained by the following observations. First, most logical topologies do not require a large number of wavelengths. Second, there are several logical topologies that are much better than any given fixed configuration but not far from the optimal configuration. Finally, there are many ways to evolve (using branch exchanges) from a given topology to the optimal topology.

If the physical topology is a bidirectional ring and the logical topology has at most one lightpath between each source and destination node pair, it can be shown that a minimum of $P(N-P)$ wavelengths are required to implement all possible logical topologies.[2,3] This lower bound on wavelength requirements is determined by constructing logical topologies that maximize the number of lightpaths routed on a given link. Simulations suggest that this lower bound on wavelength requirements is extremely tight.

Therefore, for an $N = 10$ node and $P = 2$ port network, a minimum of 16 wavelengths are required to establish all logical topologies. The impact of further restrictions on the number of network wavelengths is illustrated in Fig. 8. Reconfiguration

[2]When N is even and P is odd, and $P < (N/2)$, a lower bound of $P(N-P) - 1$ wavelengths is obtained. If N is odd and $P > (N-1/2)$, then a minimum of $(N^2 - 1/4)$ wavelengths are required. For N even and $P > (N/2)$, at least $(N^2/4)$ wavelengths are needed. For all other cases, a minimum of $P(N-P)$ wavelengths are required.

[3]In the case of $P = 1$ and N odd, the network connectivity constraint reduces the wavelength requirement from $N-1$ to $N-2$.

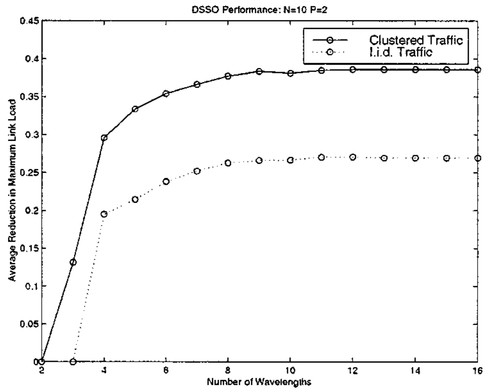


Fig. 8. The average reduction in maximum link load achieved by the DSSO reconfiguration algorithm is plotted as a function of the network wavelength restriction for dynamic traffic with $K = 20$. Results are for a network with $N = 10$ nodes and $P = 2$ ports per node.

benefits are shown for an $N = 10$ node and $P = 2$ port network as a function of the number of available wavelengths. These results indicate that most of the gain in performance can be achieved with approximately 8 wavelengths, or half the number required to establish all possible logical topologies.

VI. Conclusion

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.

The iterative reconfiguration strategy also performs well under wavelength restrictions. It was shown that the number of wavelengths needed to achieve most of the reconfiguration benefit is much less than PN, the maximum number of wavelengths necessary to realize any logical topology.

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

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.

Acknowledgment

The authors would like to thank Dr. K. Tam, Dr. S. Finn, and Dr. P. Lin for helpful discussions and suggestions.

References

[1] A. S. Acampora, "A multichannel multihop local lightwave network," in *Proc. 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] D. Bienstock and O. Gunluk, "Computational experience with a difficult mixed-integer multicommodity flow problem," *Mathematical Programming*, vol. 68, pp. 213–237, 1995.
[4] J. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Trans. Commun.*, vol. 39, pp. 1223–1230, Aug. 1991.
[5] 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.
[6] J.-F. P. Labourdette, G. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration of lightwave networks," *IEEE Trans. Commun.*, vol. 42, pp. 2822–2832, Oct. 1994.
[7] G. N. Rouskas and M. H. Ammar, "Dynamic reconfiguration in multihop wdm networks," *J. High Speed Networks*, vol. 4, no. 3, pp. 221–238, June 1995.
[8] I. Baldine and G. N. Rouskas, "Dynaic load balancing in broadcast wdm networks with tuning latenies," in *Proc. IEEE Infocom 1998*, vol. 1, pp. 78–85.
[9] ——, "Dynamic reconfiguration policies for wdm networks," in *Proc. IEEE Infocom 1999*, vol. 1, pp. 313–320.
[10] S. Banerjee and B. Mukherjee, "The photonic ring: Algorithms for optimized node arrangements," *Fiber Integrated Optics*, vol. 12, no. 2, pp. 133–171, 1993.
[11] D. Bienstock, private communication, 1999.
[12] J. A. Bannister, L. Fratta, and M. Gerla, "Optimal topologies for the wavelength-division optical network," in *Proc. EFOC/LAN 90, 8th European Fiber Optic Commun. and Local Area Networks Exposition, LAN Proc.*, 1990, pp. 53–57.
[13] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*. Englewood Cliffs, NJ: Prentic Hall-Inc., 1982.
[14] A. Narula-Tam and E. Modiano, "Dynamic load balancing for wdm-based packet networks," in *Proc. INFOCOM 2000*, pp. 1010–1019.
[15] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks," *IEEE Trans. Commun.*, vol. COM-25, 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.*, vol. 40, 1972, pp. 255–270.
[17] A. Narula-Tam, P. J. Lin, and E. Modiano, "Wavelength requirements for virtual topology reconfiguration in wdm ring networks," in *Proc. ICC*, 2000.

Aradhana Narula-Tam received the B.S.E. degree in electrical engineering from the University of Pennsylvania, Philadelphia, in 1990, and the S.M. and Ph.D. degrees, also in electrical engineering, from the Massachusetts Institute of Technology (MIT), Cambridge, in 1992 and 1997, respectively.

From 1997 to 1998 she was in the Research and Advanced Development Group at Motorola Information Systems Group, Mansfield, MA. Since September 1998 she has been with the Advanced Networking Group at MIT Lincoln Laboratory. Her research interests include communication theory and optical networks.

Eytan Modiano (S'90–M'93–SM'00) received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, in 1986, and the M.S. and Ph.D. degrees, both in electrical engineering, from the University of Maryland, College Park, in 1989 and 1992, respectively.

He was a Naval Research Laboratory Fellow between 1987 and 1992, and a National Research Council Post Doctoral Fellow during 1992–1993 while he was conducting research on security and performance issues in distributed network protocols. Between 1993 and 1999 he was with the Communications Division at MIT Lincoln Laboratory where he worked on communication protocols for satellite, wireless, and optical networks, and was the project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. Since 1999, he has been a member of the faculty of the Aeronautics and Astronautics Department and the Laboratory for Information and Decision Systems (LIDS) at MIT, where he conducts research on communication networks and protocols with emphasis on satellite and hybrid networks, and high speed networks.