# Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic

Ayşegül Gençata and Biswanath Mukherjee

*Abstract*— **We present a new approach to the virtual-topology reconfiguration problem for wavelength-routed, optical wide-area networks under dynamic traffic demand. By utilizing the measured Internet backbone traffic characteristics, an adaptation mechanism is proposed to follow the changes in traffic without assuming that the future traffic pattern is known. In that sense, our work differs from the previous studies which redesign the virtual-topology according to an expected (or known) traffic pattern, and then modify the connectivity to reach the target topology. The key idea of our approach is to adapt the underlying optical connectivity by measuring the actual traffic load on lightpaths continuously (periodically based on a measurement period) and reacting promptly to the imbalances caused by fluctuations on the traffic by adding or deleting one lightpath at a time. We aim to correct the encountered load imbalance directly, either by tearing down a lightpath that is lightly loaded or by setting up a new lightpath when congestion occurs. We introduce high and low watermark parameters on lightpath loads to detect any over/underutilized lightpath, and to trigger an adaptation step. The adaptation method is evaluated through simulations and the effect of system parameters (high and low watermarks, length of the measurement period) are investigated.**

## I. INTRODUCTION

THIS paper investigates the problem of on-line reconfiguration of the virtual-topology that arises in WDM mesh networks when the traffic load changes dynamically over time. The set of optical nodes interconnected by point-to-point fiber links constitutes the physical topology of an optical network. Lightpaths [1] are set up by configuring the optical switches and transceivers to provide single-hop communication channels between the end nodes and to eliminate the electronic processing at intermediate nodes along the path. A virtual-topology is a set of static lightpaths established to provide all-optical connectivity between nodes for a given traffic demand. However, it is usually not possible to form a fully-connected virtual-topology, because of scalability and economic issues, thereby requiring multi-hopping and electronic switching over several lightpaths for the traffic between some nodes.

The virtual-topology design problem has been studied extensively under the assumption that the traffic matrix is known a priori and is static [2], [3], [4], [5]. On the other hand, in real networks, the traffic rates between node pairs fluctuate distinguishably over time [6], [7]; an exemplary traffic measurement for a period of one day can be seen in Fig. 1 [6]. In this example, the measurements for both directions of a link are displayed as two profiles. Because traffic fluctuates, reconfiguration of the virtual-topology would be needed since a virtual-topology optimized for a specific traffic demand would lose its optimality with changes in the traffic pattern.

Reconfiguration capability of wavelength-routed networks has been emphasized as an outstanding property, and several virtual-topology reconfiguration schemes have been proposed [8], [9], [10], [11], [12]. The general approach to the virtual-topology reconfiguration problem has been a two-phase operation: first phase being a virtual-topology design for the new traffic conditions and second phase being a transition period from the old virtual-topology to the newly designed one. A major difficulty with this approach is that the future traffic demand is assumed to be known. With this information in hand, the design of a new virtual-topology was practicable, and considerable effort was given to the design of the next virtual-topology so as to reduce the changes during the transition phase [8], [11]. In practice, the assumption on the future traffic may decrease the value of the newly designed virtual-topology (and the whole reconfiguration) if the traffic changes predicted are inaccurate.

A second problem originates from the transition process. Since, during the transition phase, network resources would not be available for data transmission, several techniques to minimize the disruption to the ongoing traffic had to be developed in early studies. Changing the virtual-topology in a working network may interrupt the traffic directly: During the time period when the changes in the optical layer occur, the lightpaths involved in transition cannot be used. Previous reconfiguration studies dealt with this problem either by performing the reconfiguration on all network elements concurrently [13], or by applying step-by-step changes until the new virtual-topology is settled [9], [14]. In both cases, the traffic disruption because of the transition phase cannot be eliminated completely.

In this study, our problem definition is different: As traffic fluctuates over time, it will be monitored systematically, and no other assumption will be made on future traffic pattern. The reconfiguration process is seen as a continuous measurement-adaptation system where small adjustments are made, instead of waiting for a noticeable drop in system efficiency and changing the topology entirely. With this new definition of the problem, we expect to solve the first issue previously pointed out and to obtain a scheme which is practical for real backbone networks.

The traffic-disruption problem due to the unavailability of

This work has been supported in parts by Istanbul Technical University Grant for Long Term Research Activities and the US National Science Foundation under Grant No. ANI-98-05285.

A. Gençata is with the Department of Computer Engineering, Istanbul Technical University, Istanbul 80626, Turkey. She is a visiting Ph.D. scholar at the University of California at Davis, where this work was performed. E-mail: gencata@cs.ucdavis.edu

B. Mukherjee is with the Department of Computer Science, University of California at Davis, CA 95616, USA. E-mail: mukherje@cs.ucdavis.edu

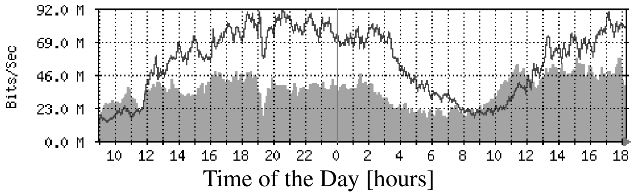


Fig. 1. Traffic measurements on a link during a 33-hour period (Abilene Network). The two profiles correspond to the two directions of traffic on a link.

network resources can also be solved by the new approach if the step adjustments on the virtual-topology are chosen carefully. In an earlier study, hitless reconfiguration was defined as a reconfiguration process without the loss of any data [10]. According to the methodology proposed in [10], the transition between topologies was achieved by first establishing all new links without removing any link. The links of the first topology were removed only when the traffic was rerouted through the links of the new topology. In our study, we design the step adjustments in a hitless manner, so that a change can be either a lightpath addition or a lightpath deletion (after rerouting the traffic using this virtual link).

Observation on several backbone networks shows that the amount of traffic between nodes changes in a smooth and continuous way [6], [7] (see Fig. 1). Therefore, the adaptation mechanism we propose would be beneficial especially for backbones since the traffic characteristics in these environments favor the slow adaptation.

In this paper, we develop a reconfiguration algorithm which applies simple adjustments to the virtual-topology when it is necessary. Network traffic is measured periodically, and the information on virtual-link loads is used to decide whether an adjustment is needed. Basically, a new lightpath is added when a congestion is encountered, and a lightpath is deleted if it is being underutilized. In our current study, only one lightpath change is allowed at the end of each observation period, although variations exist to allow multiple lightpath additions and deletions. We introduce two system parameters to detect the congestion and link-usage inefficiencies: High watermark $W_H$ and low watermark $W_L$. At the end of an observation period (typically hundreds of seconds), if the load of one or more lightpaths is higher than $W_H$, a new lightpath is established to decrease that load. When the load of a link drops below $W_L$, that link will be torn down if alternate paths exist for diverting the traffic using that lightpath.

Our scheme is computationally simple when compared to previous studies, e.g., [14], and hitless since the traffic is not directly interrupted. The system parameters $W_H$ and $W_L$ provide flexibility and control over the adaptation process, specifically the frequency of the virtual-topology adjustments, average hop distance, and the network resource usage. We elaborate on the effect of system parameters through simulations in Section IV.

In previous virtual-topology design studies, the methods established as many lightpaths as possible and reconfiguration did not change the number of lightpaths [9], [14]. Our method, on the contrary, keeps only the necessary number of lightpaths that can grow during peak traffic hours and fall when the overall network traffic decreases. This type of operation is cost effective, especially for network operators who do not own the physical infrastructure. Also, in a related study [15], we show that the adaptation method can perform reconfiguration using far fewer lightpath changes compared to an earlier reconfiguration study [11].

The rest of the paper is organized as follows: Section II explains the network and the traffic models and provides an informal statement of the problem. Selection of the best candidate lightpath to add or to drop at each step is formulated as an optimization problem in Section II-C. The key ideas of the adaptation algorithm are introduced and an outline is given in Section III. In Section IV, results from our simulation experiments are discussed. Section V concludes the paper.

## II. Problem Definition

### A. Network Model

We consider a network of N nodes connected by bidirectional optical links forming an arbitrary physical topology. Each optical link supports W wavelengths, and each node is assumed to have T transmitters and R receivers. We assume that each node is equipped with an optical cross-connect (OXC) with full wavelength-conversion capability, so that a lightpath can be established between any node pair if the resources (an optical transmitter at source, an optical receiver at destination, and at least a wavelength on each fiber link) are available along the path. Enhancing our approach to accommodate no wavelength conversion is straightforward. We consider unidirectional lightpaths, since the traffic between two nodes is not necessarily assumed to be symmetric.

Each OXC is connected to an edge device, e.g., an IP router, which can be a source or a destination of packet traffic and which can provide routing for multi-hop traffic passing by that node. We assume that each router is capable of processing all packet traffic flowing through it and of observing the amount of traffic in its outgoing lightpaths. In this paper, for ease of explanation, we consider a centralized approach to the virtual-topology reconfiguration problem. A central manager will collect the virtual-link usage information from routers at the end of every observation period. Specifically, the link-usage information needed to make a reconfiguration decision consists of only overloaded and underutilized links' usage and the end-to-end packet traffic flowing through the overloaded links. The decision for a topology change will then be made by the central manager, and a signaling mechanism will be started if a lightpath addition or deletion is required as a result of the decision algorithm (in this paper, for simplicity, we ignore the details of the signaling protocol). An implicit assumption here is that the observation period is much longer (typically hundreds of seconds or longer) than the time it takes for control signals to propagate from various nodes to the central manager.

In the optical layer, we use shortest-path routing for routing lightpaths on the physical topology and the first-fit scheme for wavelength assignment [16]. For packet routing, we consider a shortest-path (minimum-hop) routing scheme, since it provides better usage of network links and is frequently used by existing routing protocols.

### B. Traffic Model

Since this paper is focused on backbone networks, a traffic model is derived based on the observations on several backbones' link loads. Backbone traffic is the aggregation of several end systems' traffic, and the aggregation process filters out the short-term variations. On the other hand, as can be seen in Fig. 1, long-term variations (on a scale of hours) remain and repeat their pattern in one-day periods. To obtain a realistic model, we sampled some representative link traffic rates from real networks over a 24-hour period. The sampling allows us to represent the traffic rate as a function of time. These traffic-rate functions are used to generate the simulated traffic between any node pair in the network. We expect that the effect of aggregation and smoothing is more intense at the edge routers with respect to the smoothing effect of aggregation of few core node-pair traffic on a network link. Therefore, we believe that it is reasonable to use the link traffic to simulate the traffic between core routers for our purposes. This assumption is not vital for the adaptation method we propose, and the method should work for any input traffic having long-term fluctuations, since the reaction time can be adjusted according the time scale of these fluctuations. To create a simulation pattern for each node pair, a rate function is randomly selected, normalized and randomly scaled. In this model, every element of the traffic matrix is a continuous function of time representing the traffic rates during one day, rather than a single value. We should also note that no assumption on future traffic demand is being made in the adaptation scheme except that (without loss of generality) the traffic rates are expected to fluctuate slowly compared to the time interval between adaptation steps.

### C. Problem Statement

Our essential aim is to provide a good virtual-topology under dynamic traffic conditions, by keeping the lightpath loads balanced and by making changes to the virtual connectivity only when it is necessary. One can identify the following issues in such a problem:

- Traffic should be monitored continuously to provide adequate information to the reconfiguration system.
- A decision mechanism is needed to trigger a virtual-topology change, if the current topology is not convenient.
- Finally, the exact modification to the topology should be determined.

We state our approach to solve this problem formally as a mixed-integer linear program (MILP). This formulation is defined for one step of the adaptation, so it defines a local optimization method.

*Formulation of Local Optimization:* We assume full wavelength-conversion capability at each node in this formulation. We use the following notations:

- $s$ and $d$ denote *source* and *destination* of a packet when used as a superscript or subscript.
- $i$ and $j$ denote *originating* and *terminating* nodes of a lightpath, respectively.

At any step of adaptation, one of these three actions can be taken: Addition of a lightpath, deletion of a lightpath, or no change to the virtual-topology. The choice of the action is related to the highest and lowest lightpath loads, and the watermark values. Selecting the proper action and the best lightpath which keeps the maximum link load as low as possible turns out to be a local optimization problem. In the remainder of this section, we give the MILP formulation of one step of the adaptation method.

*Given:*

- Number of nodes in the network $= N$.
- Physical topology of the network $P = \{P_{mn}\}$, where $P_{mn}$ indicates the number of fibers between nodes $m$ and $n$, and $P_{mn} = P_{nm}$.
- Current traffic matrix $\Lambda = \{\Lambda_{sd}\}$ denotes the average traffic rate (in bits/s) measured during the last observation period between every node pairs, with $\Lambda_{ss} = 0$ for $s = 1, 2, ..., N$.
- Current virtual-topology $V = \{V_{ij,q}\}$, where $V_{ij,q}$ is a binary value denoting $q^{th}$ lightpath between nodes $i$ and $j$, and $V_{ii,q} = 0$. $V_{ij,0} = 0$ if there is no lightpath from node $i$ to node $j$. $V_{ij,0} = V_{ij,1} = ... = V_{ij,k-1} = 1$ and $V_{ij,k} = 0$ if there are $k$ lightpaths from node $i$ to node $j$. Since lightpaths are not necessarily assumed to be bidirectional, $V_{ij,q} = 0 \nRightarrow V_{ji,q} = 0$.
- Number of wavelengths on each fiber $= W$.
- Capacity of each wavelength channel $= C$ (in bits/s).
- Number of transmitters and receivers at node $i$: $T_i$ and $R_j$ respectively.
- High watermark value $= W_H$.
- Low watermark value $= W_L$.
- Highest and lowest lightpath loads measured during the observation period: $L^P_{Max}$ and $L^P_{Min}$, respectively (in bits/s).

*Variables:*

- Physical routing binary variable $p^{ij,q}_{mn} = 1$ if the $q^{th}$ lightpath from node $i$ to node $j$ is routed through the physical link $(m, n)$.
- New virtual-topology: $V' = \{V'_{ij,q}\}$, $V'_{ij,q}$ is defined similar to $V_{ij,q}$. Note that $V'$ is at most one lightpath different from $V$.
- Traffic routing: The binary variable $\Upsilon^{sd}_{ij,q}$ is 1 when the traffic flowing from node $s$ to node $d$ employs $V'_{ij,q}$ as an intermediate lightpath, and 0 otherwise. $\Upsilon^{sd}_{ii,q} = 0$ by definition. The traffic from $s$ to $d$ is not bifurcated, i.e., any packet between $s$ and $d$ will flow through one path.
- Load of maximally-loaded lightpath in the network: $L_{Max}$.

*Objective:*

$$\text{Minimize} \quad L_{Max}$$

The objective function minimizes the load of the maximally-loaded lightpath in the network. This objective allows us to balance the network load in the new virtual-topology, by addition or deletion of the best possible lightpath.

*Constraints:*

- On physical topology:

$$\forall i, j, q, \quad \sum_n p^{ij,q}_{in} = V'_{ij,q} \tag{1}$$

$$\forall i, j, q, \quad \sum_n p_{nj}^{ij,q} = V'_{ij,q} \tag{2}$$

$$\forall i, j, q, l, \quad \sum_n p_{nl}^{ij,q} - \sum_n p_{ln}^{ij,q} = 0 \tag{3}$$

$$\forall m, n, \quad \sum_i \sum_j \sum_q p_{mn}^{ij,q} \leq W \cdot P_{mn} \tag{4}$$

$$\forall m, n, i, j, q, \quad p_{mn}^{ij,q} \leq V'_{ij,q} \tag{5}$$

Equation (1) ensures that only one outgoing physical link of the source node will be assigned to a lightpath. Equation (2) ensures that only one incoming physical link of the destination node will be assigned to a lightpath. Equation (3) guarantees that the number of incoming and outgoing links reserved for a lightpath at any intermediate node will be equal. Number of wavelengths are limited to $W$ on every physical link by Equation (4). Equation (5) states that a physical link is assigned only if the lightpath exists.

- On virtual-topology connections:

$$\sum_i \sum_j \sum_q V'_{ij,q} = \sum_i \sum_j \sum_q V_{ij,q} + k_H - (1 - k_H) \cdot k_L \tag{6}$$

where

$$k_H = \left\lceil \frac{L_{Max}^P}{C} - W_H \right\rceil \text{ and } k_L = \left\lceil W_L - \frac{L_{Min}^P}{C} \right\rceil$$

$$\forall i, j, q, \ \left[1 + 2 \cdot (k_H - 1) \cdot k_L\right] \cdot (V'_{ij,q} - V_{ij,q}) \geq 0 \tag{7}$$

The values of $k_H$ and $k_L$ are calculated by using maximum and minimum lightpath loads measured in the last observation period, watermark values, and channel capacity. Therefore, these values are constant for the MILP. Note that $k_H$ is unity when one or more lightpaths are experiencing heavy load. That will ensure that a new lightpath will be added to the virtual-topology. $k_L$ is unity when one or more lightpaths has a load below the low watermark, and in this case a lightpath will be deleted if $k_H = 0$. Equation (6) specifies the total number of lightpaths in the new virtual-topology, i.e., it decides whether a change should be made, and if the answer is affirmative, whether a lightpath should be added or deleted. Equation (7) guarantees that the new virtual-topology is constituted by the same set of lightpaths of the current virtual-topology except one lightpath being added or deleted.

- On virtual-topology traffic variables:

$$\forall s, d, l, q \ \sum_i \Upsilon_{il,q}^{sd} - \sum_i \Upsilon_{li,q}^{sd} = \begin{cases} 1 & l = d \\ 0 & l \neq s \text{ and } l \neq d \\ -1 & l = s \end{cases} \tag{8}$$

$$\forall s, d, i, j, q \quad \Upsilon_{ij,q}^{sd} \leq V'_{ij,q} \tag{9}$$

$$L_{Max} \leq C \cdot W_H \tag{10}$$

$$\forall i, j, q \quad \sum_s \sum_d \Lambda_{sd} \cdot \Upsilon_{ij,q}^{sd} \leq L_{Max} \tag{11}$$

Equation (8) is a multicommodity-flow equation controlling the routing of packet traffic on virtual links. Equation (9) ensures that traffic can only flow through an existing lightpath, and Equation (10) specifies the capacity constraint for any lightpath. Equation (11) provides that the load on any lightpath is smaller than or equal to the maximum load $L_{max}$.

- On transceivers:

$$\forall i \quad \sum_j \sum_q V'_{ij,q} \leq T_i \tag{12}$$

$$\forall j \quad \sum_i \sum_q V'_{ij,q} \leq R_j \tag{13}$$

Equations (12)–(13) limit the total number of lightpaths originating from and terminated at a node to the total number of transmitters and receivers at that node.

The above formulation gives the best selection for a virtual-topology adjustment of one lightpath. A comparison of our adaptation scheme by solving the MILP to an earlier reconfiguration method in [11] can be found in [15]. Since the time complexity of the MILP limits its use for large networks, we turn to heuristic methods.

## III. Heuristic Adaptation Algorithm

### A. Key Ideas

We propose a one-phase reconfiguration algorithm where the network traffic flows are continuously observed and simple updates to the virtual-topology are made when it is necessary. The key idea of our approach is to set up new lightpaths when congestion occurs or tear down the existing lightpaths when they are not used efficiently. Since frequent changes to the virtual-topology are not desirable, selection of traffic-observation periods and decisions of triggering a virtual-topology change (in this case, either an addition or a deletion of a lightpath) should be made carefully.

Length of the observation period is a parameter which can be selected and updated according to the specific properties of the network traffic. It specifies in what frequency the adaptation algorithm is activated. We use load-balancing ability of the current topology as an input to the triggering function of the adaptation algorithm. Since, having perfectly balanced loads on every link is practically not possible, we aim to maintain quasi-balanced virtual links. We can define a quasi-balanced topology as follows:

$$V = \{V_1, V_2, \ldots, V_k\}$$
$$W_L \leq \delta(V_i) \leq W_H \quad i = 1, 2, \ldots, k$$

Here, $\delta(V_i)$ is the load of the $i^{th}$ lightpath, and $W_L$ and $W_H$ are the low watermark and the high watermark values. The objective of the adaptation algorithm is to keep the load of every lightpath between these two thresholds. At the end of an observation period, if the load of one or more lightpaths is higher than $W_H$, the lightpath $V_{max}$ having the maximum load $\delta_{max} = \delta(V_{max})$ will be considered, and a new lightpath will be established to decrease $\delta_{max}$. The new lightpath $V_{new}$ should carry at least one of the traffic flows that $V_{max}$ is carrying, so that, after the new lightpath is established, the load of $V_{max}$ can decrease. As an extreme case, one may set $W_H = 100$ and $W_L = 0$ minimizing the changes on the virtual-topology but also decreasing the quick reaction ability of the network to dynamic changes in the traffic. The characteristic of the traffic should determine the proper choice for the watermark values.

There are several ways to select the source and the destination nodes of the new lightpath to be added. We choose to establish a lightpath between the end nodes of the multi-hop traffic with the highest load using $V_{max}$. Doing so, we expect to have a noticeable decrease in $\delta_{max}$. If $V_{max}$ is only carrying single-hop traffic, then we simply add a new lightpath between the end nodes of $V_{max}$. In that case, the load of $V_{max}$ will decrease to $\delta_{max} \cdot \frac{|V_{max}|}{|V_{max}|+1}$.

When the load of a virtual link decreases to a value below $W_L$, either because of changes in the dynamic traffic or rerouting as a result of previous topology changes, the algorithm will tear down that link, unless it is on a critical path for a traffic flow. The main idea behind the deletion of underloaded lightpaths is to release the network resources whenever they are not used efficiently and to have them available when new lightpaths are to be established. In some cases, it is possible for the deletion of a lightpath to lead to congestion on another lightpath and consequently to an addition of a new lightpath. However, the traffic rates are changing dynamically, and in such an environment that behavior should not be interpreted as instability. This dynamism is necessary for the method to find a good set of lightpaths for the current traffic. Whenever required, this ping-pong effect can be controlled by adjusting the watermark values.

Whenever a lightpath is added or deleted, the routes for all packet traffic is re-calculated. Occasionally, new congestions may occur after rerouting due to the change of the routes of some traffic, but this is a part of the adaptation process and indicates that more lightpaths should be added or deleted. In some cases, a few more steps may be necessary for the virtual-topology to reach a load-balanced state.

We should also note that variations to our approach exist, such as deletion or addition of multiple lightpaths in one step when the load of more than one lightpath is outside the region defined by watermarks. In this paper, we study the basic case where only one lightpath is allowed to be added or deleted, and several other variations need to be studied in a future work.

### B. Outline

We give a pseudo-code of the heuristic adaptation algorithm in Fig. 2. We use $\lambda_{sd}$ to denote the traffic rate from node $s$ to node $d$ normalized to the capacity of a virtual link. The notation $(i, j)$ is used to express the originating and the terminating nodes of a lightpath, and $G$ to represent the virtual-topology graph. $G - V_{ij}$ is a subgraph of G with one edge from node $i$ to node $j$ extracted.

The algorithm first checks for any lost end-to-end traffic because of a lack of connectivity in the virtual-topology. Under normal operation, the adaptation process expects a connected network of already established lightpaths and it preserves the connectivity of node pairs as long as these nodes have a nonzero rate of traffic flowing between them. In some cases, traffic rate from a node $s$ to a node $d$ may cease, and if this silent period is longer than an observation period, a lightpath providing the connectivity may be torn down.

---

**I**nput:

- Virtual-topology, $V = \{V_{ij}\}$
- Current traffic rates, $\lambda_{sd}$
- Load of virtual links, $\delta_{ij}$
- Available number of transmitters at each node, $T_i$ and $R_i$

**O**utput:

A decision which can be one of the followings:

- Addition of a lightpath between $i_{new}$ and $j_{new}$
- Deletion of an existing lightpath between $i_d$ and $j_d$
- No change to the virtual-topology

**A**lgorithm:

At the end of every observation period:

```
If λ_sd > 0 and there is no path from s to d then
    Establish V_sd for max_{sd,V_sd=0}{λ_sd}
else
    Find the maximum link load: δ_max(i_max, j_max)
    if δ_max > W_H then
        max-load ← 0
        for every multi-hop traffic flow λ_sd using
                                lightpath (i_max, j_max) do
            if (λ_sd > max-load) and (T_s > 0) and
                                        (R_d > 0) then
                max-load ← λ_sd
                s_mm ← s
                d_mm ← d
            endif
        endfor
        if max-load > 0 then
            Establish V_{s_mm d_mm}
        else
            Establish V_{i_max j_max}
        endif
    endif
    if no lightpath can be added then
        Find the minimum link load: δ_min(i_min, j_min)
        complete ← false
        while (δ_min < W_L) and not complete do
            if (i_min, j_min are connected in
                        G − V_{i_min j_min}) or (δ_min = 0) then
                Tear down V_{i_min j_min}
                complete ← true
            else
                Find the next lowest link load: δ_nextlow
                δ_min ← δ_nextlow
            endif
        endwhile
    endif
endif
```

---

Fig. 2. Pseudo-code of heuristic adaptation algorithm.

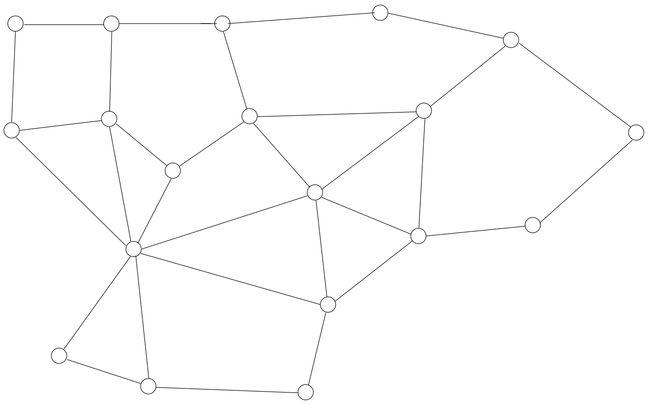

Fig. 3. An example telco network used as the physical topology in our study.

If all of the communicating node pairs have at least a route, then the link loads are considered next. A change to the virtual-topology is made only when a link load is higher than $W_H$ or lower than $W_L$; otherwise, no change is made until the end of the next observation period. The maximum link load is considered first, since a highly-loaded link (and with a potentially increasing load) would cause more serious problems compared to an underutilized link. Lightpath deletion is performed only if that link is not on the unique path between a node pair $i$ and $j$ for which the traffic rate $\lambda_{ij} > 0$. All lightpaths with a load lower than $W_L$ are considered in increasing order of their loads, until an appropriate lightpath is found.

## IV. Performance Studies

### A. Simulation Environment

Simulation experiments were conducted to investigate the fitness of the adaptation approach and to expose the effect of system parameters. In our experiments, we implemented the heuristic algorithm shown in Fig. 2. As the physical topology, a telco mesh network of 19 nodes interconnected by 31 bidirectional links was used (Fig. 3). We assume that the number of wavelengths is 16 per fiber link and equal for all links. Each node is assumed to have 8 transmitters and 8 receivers. The traffic matrix was randomly generated according to the model explained in Section II-B, having nonzero rate functions for 60% of the node pairs (values on the main diagonal remain zero). We use a scaling factor $\alpha_{sd}$ to create discrepancies in volume of the traffic flowing between different node pairs where $\alpha_{sd}$ takes random values in the range $[0.2, 1.2]$. (A traffic pattern chosen for a node pair $< s, d >$ is multiplied by the factor $\alpha_{sd}$.) As a result, traffic rates were up to the $15\%$ of the capacity of a lightpath, creating a sufficient overall network load that is of interest in demonstrating the effects of the various system parameters. The traffic rate at a point in time between two sampling points of the rate function was calculated by a linear interpolation to represent the continuous change in traffic.

In this study, we employ the following performance metrics to reveal the different aspects of the adaptation scheme: The average time between consecutive reconfiguration steps, the traffic weighted average hop distance, transceiver usage all over

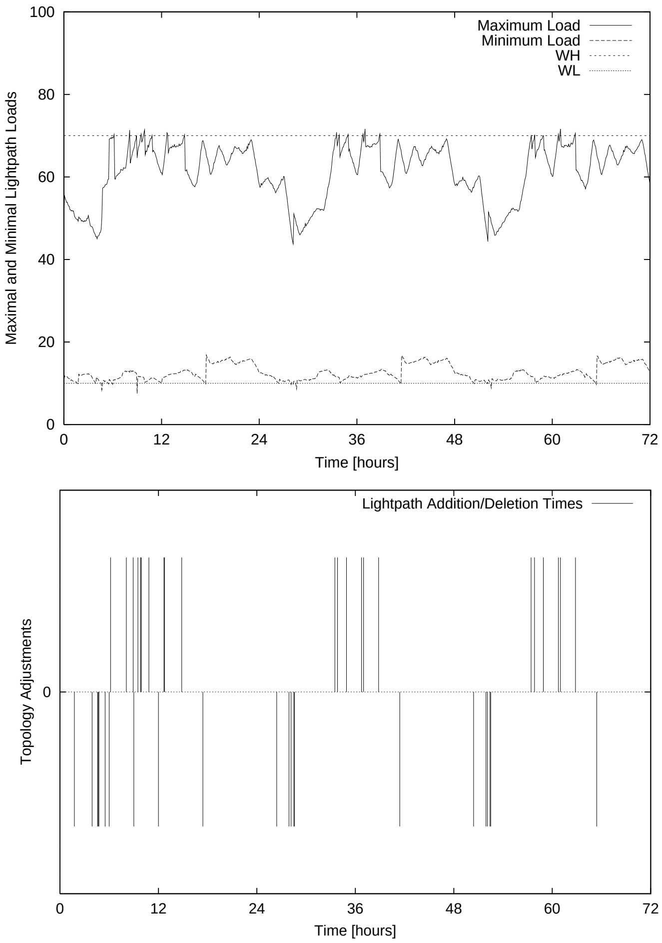


Fig. 4. a. Maximal and minimal lightpath loads in network during a 3-day run ($W_H = 70, W_L = 10$). b. Impulse graphic indicating times of lightpath addition or deletion.

the network, and the percentage of link loads being in the balanced region $[W_L, W_H]$. Through these metrics, we examine the effects of changing the system parameters, specifically high watermark, low watermark, and length of observation period.

### B. Typical Operation

As a primary experiment, we demonstrate the operation of the system by measuring the maximal and minimal lightpath loads in the network at the end of every observation period. High and low watermarks will define a *balance region* in which the lightpath loads are allowed to fluctuate as long as they do not exceed the high watermark or drop below the low watermark. Every time a link load goes out of the balance region, the system will try to adapt to the new traffic conditions.

An exemplary operation of the system for a 3-day period with $W_H = 70$ and $W_L = 10$ is shown in Fig. 4. Figure 4(a) plots the maximal and minimal loads in network observed at the end of each observation period (300 sec for this experiment). Figure 4(b) shows the times of topology adjustments, where a positive impulse indicates a lightpath addition and a negative impulse indicates a lightpath deletion. The same average traffic rate functions were repeated for every day, but slight differences can be seen for the same time of different days related to randomly generated traffic and to the dynamic nature of

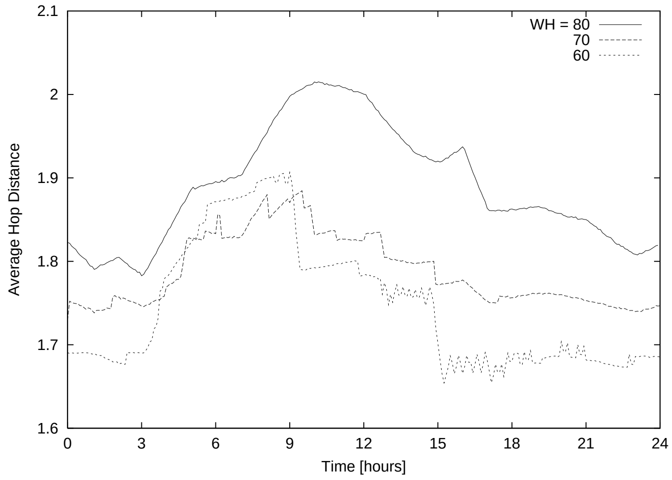


Fig. 5. Traffic-weighted average hop distances during a day, for different values of $W_H$ with $W_L = 10$.

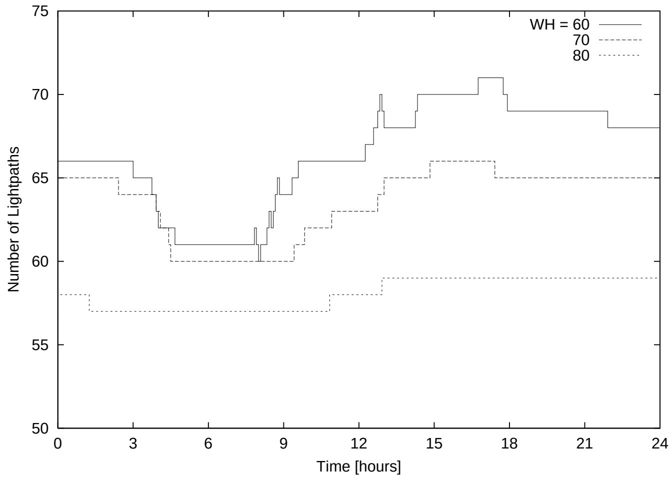


Fig. 7. Number of lightpaths during a day, for different values of $W_H$ with $W_L = 10$.

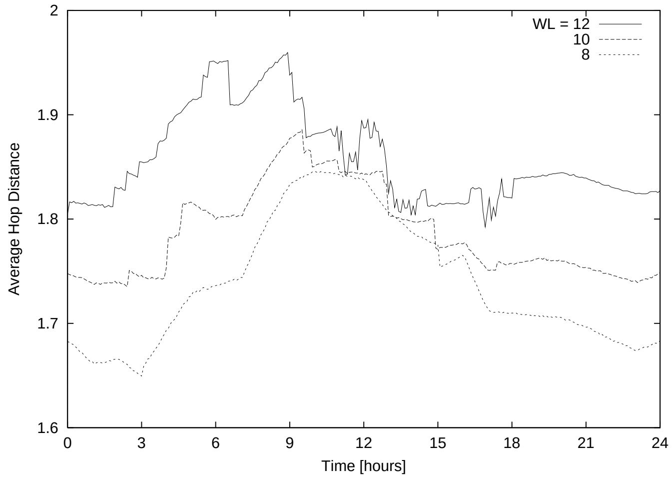


Fig. 6. Traffic-weighted average hop distances during a day, for different values of $W_L$ with $W_H = 70$.

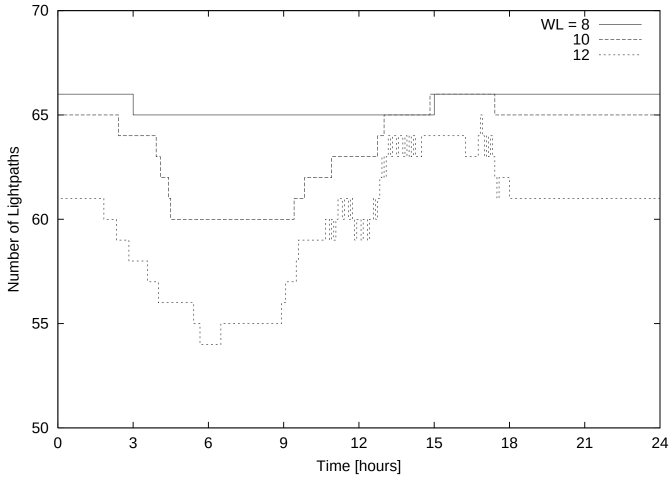


Fig. 8. Number of lightpaths during a day, for different values of $W_L$ with $W_H = 70$.

the algorithm. One may propose a scheme where strict virtual-topologies are designed for specific hours of the day and used in sequence, but that approach would be restrictive since the traffic rates in a network may change patterns also in weekends, holidays, etc. The dynamism of the algorithm provides flexibility against future traffic behavior.

### C. Effect of Altering Watermarks

The effect of changing the value of $W_H$ on traffic-weighted average hop distance while $W_L$ remains constant is shown in Fig. 5. The hours on the x axis correspond to the real hours of the day. During the night (until 9 or 10 in the morning), as the traffic load decreases all over the network, the lightpath loads decrease as well and the adaptation scheme deletes the lightly-loaded links causing an increase in average hop distance.

Reciprocally, the average hop distance decreases with increasing traffic load, and this observation is valid in general, i.e., for any value of $W_H$ and $W_L$. As can be seen in the figure, the average hop distance decreases when $W_H$ takes smaller values, since the lightpaths are allowed to carry less load and the $W_H$ limit can be reached more quickly, resulting in new lightpath additions. For the case where $W_H = 60$ and $W_L = 10$, more lightpaths are deleted compared to the case where $W_H = 70$ during nighttime. This effect manifests itself in Fig. 5 as an aggressive climb in the average hop distance, and is related to the tightness of the watermarks: As the values of the watermarks approach each other, more lightpaths have loads closer to the bounds, and small changes in traffic may result in topology changes.

Figure 6 plots the effect of $W_L$ on the average hop distance when $W_H$ is held constant. When $W_L$ is small, lightpaths are deleted less frequently since fewer lightpaths will have their load drop below a small $W_L$, compared to the case with a higher $W_L$ value. Therefore, the virtual-topology keeps more lightpaths alive and has a lower hop-distance value.

The total number of lightpaths in the virtual-topology during a one-day period is plotted in Fig. 7 for different values of $W_H$. Higher values of $W_H$ result in smaller number of lightpaths and more economic usage of network resources such as transmitters. Consistent with the results obtained for average hop distance, the number of lightpaths is smaller when the network load decreases due to time-of-the-day effect.

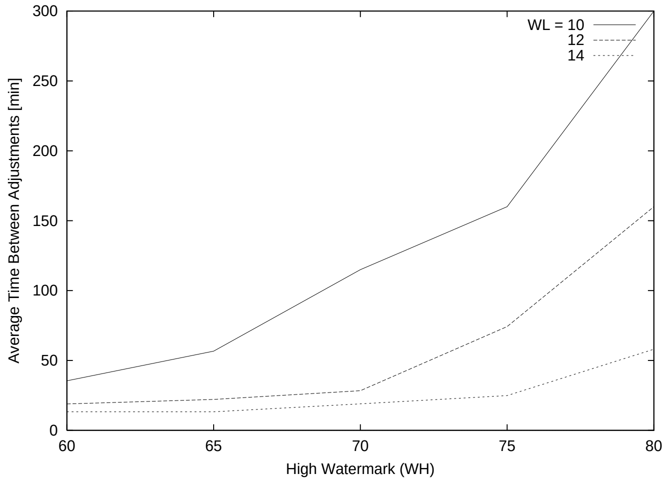


Fig. 9. Average time between two consecutive reconfiguration steps, as a function of $W_H$.

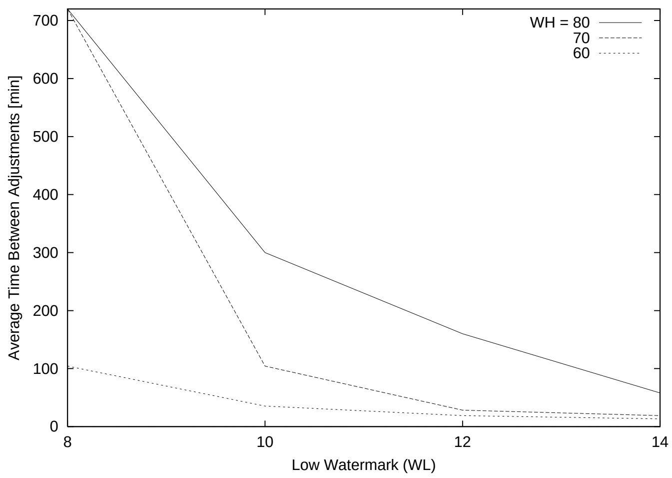


Fig. 10. Average time between two consecutive reconfiguration steps, as a function of $W_L$.

Figure 8 plots the number of lightpaths for different values of $W_L$. The number of lightpaths in the system decreases when $W_L$ increases because the adaptation scheme tends to delete lightpaths more aggressively. The total number of lightpaths is smaller when $W_L$ takes higher values. Since a larger number of lightpaths requires larger amount of network resources, one may prefer to keep this number small for economic use of resources. The two system parameters $W_H$ and $W_L$ can be used to balance the use of resources and the average hop distance.

Figures 9 and 10 plot the average time between two consecutive adjustments in virtual-topology, as a function of $W_H$ and $W_L$, respectively. We observe in Fig. 9 that the virtual-topology is adjusted less frequently when $W_H$ takes higher values. Similarly, when $W_L$ takes higher values, the virtual-topology is adjusted more frequently as can be seen in Figure 10. These two figures show that the average number of lightpath additions and deletions can be controlled by appropriately choosing the watermarks.

### D. Effect of Observation Period

A third parameter that affects the system's behaviour is the length of observation period over which the lightpath loads are

TABLE I
PROBABILITY OF ADJUSTMENT DECISION AT THE END OF AN OBSERVATION PERIOD ($W_H = 70, W_L = 10$)

| Length of Observation Period [sec] | Total Number of Periods in One Day | Average Number of Adjustments | Adjustment Probability |
|---|---|---|---|
| 100 | 864 | 12.33 | 0.014 |
| 200 | 432 | 12.17 | 0.028 |
| 400 | 216 | 14 | 0.065 |
| 800 | 108 | 17.6 | 0.163 |

measured. The average traffic rate on a lightpath is calculated by measuring the total amount of traffic passing over it and dividing that amount by the period. The lightpath load is the ratio of the measured average traffic rate to the lightpath capacity. Since the traffic rate is taken as an average value during a period, the length of one period will affect the system's ability to track the changes in traffic. Momentary changes could be measured and reacted to by using enough short observation periods, but a smoothing effect on traffic measurements by using longer periods would be more accurate to avoid the instabilities and frequent adjustments. On the other hand, long periods would decrease the efficiency of step-by-step adaptation on following the traffic changes closely since we do not allow more than one lightpath to be changed at the end of each period. Table I shows the effect of the observation-period length on topology adjustment decisions. In this experiment, $W_H = 70$ and $W_L = 10$, and the number of adjustments in a day is averaged over a 5-day run.

The first column shows the length of observation period in seconds, used during the experiments. The values in the second column are the total number of observation periods in a day; therefore, they also show how many times the adaptation algorithm was activated. The average number of adjustments during one day (number of additions and number of deletions are almost equal) is shown in the third column. The probability of adjustment at the end of an observation period is the ratio of the value in the third column to the one in the second column. As observation period gets longer, it is more likely that the algorithm will decide to make an adjustment in the virtual-topology. This result confirms our claim above; short observation periods keep track of the traffic more closely.

### E. Load Balancing

The adaptation mechanism works to balance the lightpath loads all over the network to maintain all loads in the balance region. In this section, we show the load-balancing ability of the system. Figure 11 plots the distribution of lightpaths according to their loads, for three different watermark value pairs. At the end of each observation period (300 sec. for this example), the loads of all lightpaths in the network are measured and the occurrence of each load is added to the total occurrences of that load since the beginning of the simulation. This process is repeated at the end of every observation period throughout the simulation. A point in this graphic indicates the percentage of the lightpaths with that load in the course of the experiment

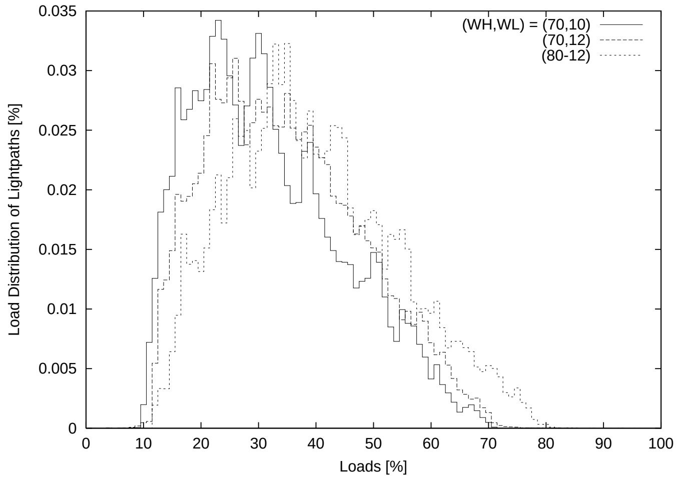


Fig. 11. Distribution of lightpath loads for different values of watermarks over the course of the simulated five days.

TABLE II
PERCENTAGE OF THE LIGHTPATH LOADS BETWEEN THE WATERMARKS OVER THE COURSE OF SIMULATION.

| | $W_L$ | | | |
|---|---|---|---|---|
| $W_H$ | 8 | 10 | 12 | 14 |
| 60 | 99.9976 | 99.9958 | 99.9932 | 99.9882 |
| 65 | 99.9995 | 99.9986 | 99.9964 | 99.9919 |
| 70 | 99.9999 | 99.9991 | 99.9971 | 99.9949 |
| 75 | 99.9999 | 99.9991 | 99.9987 | 99.9962 |
| 80 | 99.9999 | 99.9997 | 99.9994 | 99.9984 |

(a 5-day run). The figure shows that a high percentage of the links are in the balance region. Another result is that most of the lightpaths are gathered in a smaller region toward the middle of the balance-region, showing that few lightpaths are critically close to the watermarks, and they can trigger a topology adjustment in the following observation period. The same experiment is repeated for different values of $W_H$ and $W_L$, and the percentage of the lightpaths in the balance region is tabulated in Table II. As $W_H$ and $W_L$ are relaxed, the percentage of lightpaths in the balance region increases. This table shows that the watermarks can be adjusted to obtain a more/less aggressive topology adaptation.

## V. CONCLUSIONS

Based on a new view of the virtual-topology reconfiguration problem, we presented an adaptation scheme for WDM mesh networks under dynamic traffic. We defined the problem as following the daily traffic fluctuations by adapting the topology in a measurement-adaptation cycle with the following constraints: No assumption should be made on future traffic rates, and the ongoing traffic should not be interrupted by a transition phase. Our work is motivated by the traffic characteristics of backbone networks, which are changing slowly and according to the time-of-the-day.

We developed an algorithm which adjusts the virtual-topology by adding or deleting one lightpath at the end of a measurement cycle if it is necessary to do so. We introduced two system parameters, $W_H$ and $W_L$, on lightpath loads to detect any over/underutilized lightpath. Through simulations, we showed that these two parameters can be used to regulate the functioning of the proposed topology-adaptation method. We also presented the integer linear program (ILP) formulation for selecting the lightpath to be added or deleted to minimize the maximum load in network for an adaptation algorithm which allows one lightpath change at a time.

We believe that this work brings the virtual-topology reconfiguration problem from a perspective where it is considered as an interrupted process, to a new one where it is taken as an evolutionary process. A future direction is to investigate how high and low watermarks can be changed dynamically to satisfy specific performance demands. The algorithm we introduced can be enhanced in different directions: Application of new heuristics to our scheme, to select the lightpath to be added or deleted, is possible and the performance of different heuristics can be evaluated using the mixed-integer linear program formulation presented in this paper. Variations of the algorithm allowing multiple lightpath additions and deletions at each step can be studied. The development of a distributed version of the algorithm also needs further study.

## REFERENCES


[1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171–1182, July 1992.

[2] B. Mukherjee, *Optical Communication Networks*, McGraw-Hill, July 1997.

[3] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 684–696, Oct. 1996.

[4] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840–851, June 1996.

[5] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 186–198, Apr. 2001.

[6] "Abilene network traffic statistics, http://www.abilene.iu.edu/," .

[7] "Cooperative association for Internet data analysis, http://www.caida.org," .

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

[9] 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.

[10] K. Bala et al., "Toward hitless reconfiguration in WDM optical networks for ATM transport," in *Proc. of IEEE GLOBECOM*, 1996, pp. 316–320.

[11] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 598–607, Oct. 2000.

[12] B. Ramamurthy and A. Ramakrishnan, "Virtual topology reconfiguration of wavelength routed optical WDM networks," in *Proceedings of IEEE GLOBECOM*, 2000, pp. 1269–1275.

[13] 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, 1995.

[14] A. Narula-Tam and E. Modiano, "Dynamic load balancing for WDM based packet networks," in *Proc. of IEEE INFOCOM*, 2000, pp. 1010–1019.

[15] A. Gençata, L. Sahasrabuddhe, and B. Mukherjee, "Virtual-topology adaptation with minimal lightpath change for dynamic traffic in WDM mesh networks," in *Proceedings of OFC*, 2002.

[16] H. Zang and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, Jan. 2000.